

A MIXED INTEGER LINEAR PROGRAMMING APPROACH
TO FOREST UTILISATION MANAGEMENT PROBLEMS

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy in Forestry
in the
University of Canterbury
by

L. R. Broad

University of Canterbury

1985

FORESTRY

THESIS

SD
541
.B863
1985

i

In memory of Bruce Telford Drummond

ACKNOWLEDGEMENTS

Especial thanks to Professor McKelvey for providing both encouragement and leadership, skills he used so constructively to effect the running of the Forestry School during the period he held the chair. Thanks are also due to Dr. Whyte, my supervisor, without whose help this study would neither have been instigated nor completed, his assistance and advice are gratefully acknowledged; to Dr. George (Operations Research) for making time available for discussion. Assistance was also given by Dr. Walker in allowing me to read his yet unpublished book titled "How to recognise educational paradigms".

Thanks are due to Dr. Robinson (Mathematics) for his work on multi-stage processes which served as the basis for Chapter 4; to Professor Deely (Statistics) for a rhapsodical introduction to decision theory (a subject that engenders indecision), and his colleagues Drs Chacko, Smith and Wood for presenting the rudiments of statistical theory in a thoroughly understandable way, i.e., statistics without sadistics.

Many people deserve mention for their tolerance and forbearance amongst whom are my typist, Mary Kinniard, and flatmates. The last bracket is inclusive of Master Ritchie whose culinary skills with the genus *Cucurbita* must be mentioned.

The author gratefully acknowledges financial support received from the T.W. Adams scholarship.

ABSTRACT

The research reported here concerns the use of mathematical programming techniques to model resource flows in a system comprising industrial forests and subsequent wood processing and marketing activities. A review is made initially of methods to delineate management alternatives in industrial forests through formulating and solving linear programs. Linear programming techniques used to represent the associated Forest Management Problem (FMP) are discussed and solution methods analysed. The use of linear programming and mixed integer linear programming to represent resource flows in a problem where, in addition to forest management activities, the utilisation and marketing of wood based resources are also considered is explored in considerably more depth. Previous research on this latter class of problem, termed a Forest Utilisation Management Problem (FUMP), has been limited. Forest utilisation management problems may be characterised by the joint occurrence of standard Operations Research problems such as those of location, resource allocation, budget measures, and fixed charge specification. Mixed integer linear programming techniques appeared to provide a viable means to resolving FUMPs that pose non-convex programming problems. The possibility of redundancy in FUMPs is considered, and a technique to assist in preventing it is presented. The implications of redundancy to problem formulation are made apparent. Discussion also covers representational difficulties anticipated in certain components of FUMPs. A small test problem is discussed in relation to data requirements, matrix

generation, and report writing. Recommendations concerning formulation and solution of FUMPs are made. Conclusions drawn relate to the feasibility of representing FUMPs as a class of mathematical program.

Key Words : Forest planning model, wood utilisation, mixed integer linear programming, mathematical programming.

CONTENTS

<i>Chapter</i>		<i>Page</i>
	Dedication	i
	Acknowledgements	ii
	Abstract	iii
1	STUDY OUTLINE	1
	1.1 Study objectives	7
	1.2 Outline of chapter contents in thesis	8
2	FOREST REGULATORY MECHANISMS	10
	2.1 Stand level management mechanisms	11
	2.2 Forest level management mechanisms	16
	2.2.1 Representation of management units	18
	2.2.2 Single crop-type structural constraints	19
	2.2.3 Single crop-type regulatory constraints	22
	2.2.4 Multiple crop-type structural constraints	23
	2.2.5 Multiple crop-type regulatory constraints	29
	2.3 Bounding and smoothing regulatory constraints	31
	2.4 Solution techniques for forest management problems	37
	2.4.1 Standard simplex methods	38
	2.4.2 Decomposition methods, model I	39
	2.4.3 Decomposition methods, model II	41
	2.4.4 Dual variable estimation	42
	2.5 Regulatory constraints in FUMPs	46

<i>Chapter</i>		<i>Page</i>
3	TRANSPORT UTILISATION AND MARKETING	48
	3.1 Description and measurement of resource flows	53
	3.2 The transportation mechanism	58
	3.3 Introductory regulatory mechanisms	59
	3.3.1 Non-convex interpolating binary mechanism	62
	3.3.1.1 The location problem	64
	3.3.1.2 The resource allocation problem	65
	3.3.1.3 The budgeting problem	71
	3.3.1.4 The fixed charge problem	74
	3.3.2 Non-convex non-interpolating binary mechanism	76
	3.3.3 Convex mechanisms	
	3.3.3.1 Constant returns to scale	78
	3.3.3.2 Diminishing returns to scale	80
	3.4 Marketing mechanisms	84
	3.5 Structural considerations for FUMPs	88
	3.5.1 Bender's algorithm	88
	3.5.2 Processing networks	91
4	SPATIAL AND TEMPORAL EFFECTS OF ROUNDWOOD AVAILABILITY	94
	4.1 The resource digraph approach	96
	4.2 The manufacturing digraph approach	103
	4.3 The generator set approach	115
	4.4 Implementation of set mapping procedures	119

<i>Chapter</i>		<i>Page</i>
5	COMPUTATIONAL EXPERIENCE	121
5.1	Description of test problem	122
5.1.1	Data base requirements	129
5.1.2	Report writer output description	144
5.1.3	Singularity report	150
5.2	Matrix generation: development and validation	153
5.2.1	Observations on matrix generators	156
5.2.2	Observations on report writers	160
5.3	Restricted answers from planning models	162
5.3.1	Model aggregation	163
5.3.2	Estimation errors	164
5.3.3	Stochastic elements	165
5.3.4	Cost considerations	166
5.3.5	Uncertainty in economic efficiency measures	167
6	DISCUSSION AND CONCLUSIONS	169
6.1	Discussion	169
6.2	Study conclusions and recommendations	179
	REFERENCES	183
	APPENDICES	190

LIST OF FIGURES

<i>Figure</i>		<i>Page</i>
2.1	Alternatives for a model I management unit	20
2.2	Digraph showing model I management alternatives	25
2.3	Digraph showing model II management alternatives	27
2.4	A resource lower bound envelope	34
2.5	A resource upper bound envelope	36
3.1	General flow digraph	51
3.2	Control of continuous weights	63
3.3	Introduction cost vs capacity	81
3.4	Process types in a processing network	92
4.1	Resource digraph for a four-stage process	97
4.2	Manufacturing digraph representing a three-stage process	104
4.3	Resource digraph representing a three-stage process	106
5.1	General flow digraph illustrating option on continued log sales, and expansion of processing and marketing options	127
5.2	Generalised growth curves in even-aged stands	134
5.3	Cutting patterns for two type I management units	137
5.4	Generalised roundwood conversion subdigraph modelled as a three-stage conversion process	142

LIST OF TABLES

<i>Table</i>	<i>Page</i>
2.1 Tableau of crop-type revenues and costs	13
3.1 Tabulation of resource bounds	68
3.2 Tabulation of repayment sequences	73
4.1 The forward-backward mapping procedure	102

CHAPTER 1

STUDY OUTLINE

To specify a single *raison d'être* for all integrated forest products companies, other than economical survival, is difficult in that such companies may differ markedly from each other in their management policies. What can be said, however, is that all integrated forest products companies face in common major problems of resource allocation in their operational management, and that these problems arise naturally as part of the company's activities.

Problems encountered involve scheduling

1. Roundwood harvested from forest estates and other supply sources regulated by the company;
2. Processing activities where utilisation of roundwood takes place;
3. Product sales at markets;
4. Transportation of roundwood or roundwood based products; and
5. Finance, energy and manpower requirements.

Each of these problems is basic to the activities of an integrated forest products company, and as discussed later in this chapter, solutions appropriate to all of these problems are not usually obtained by attempting solutions to individual problems.

The problems listed above for an integrated forest products company illustrate component problems of what shall be termed a Forest Utilisation Management Problem (FUMP) in this study. A FUMP is, as its name suggests, concerned with the growing and the utilisation of forest resources, and

the management problems posed by integrating these two aspects. If FUMPs are considered at regional or national level, their regional or sector planning problems would rise (Baird and Whyte, 1982; Whyte, 1984). Related to a Forest Utilisation Management Problem is the notion of a Forest Management Problem (FMP). FMPs are concerned with the growing of forest resources only, and with the management problems posed therein. Explicit consideration is not given to utilisation alternatives. However, implicit consideration may be given in that, for example, a management decision may be to determine which roundwood resources to produce; these resources being strongly linked to their intended use within some form of utilisation. Typically, mechanisms to regulate forest growth can be considered to arise from FMPs. The FMPs of interest to this study will be those that give rise to linear programming formulations (Johnson and Scheurman, 1977). To facilitate discussion the distinction between an FMP and the forms of representation it gives rise to will often be dropped, similarly for FUMPs.

To conduct an examination of the resource allocation problems of a FUMP, note the distinct notions of

1. A *production process* (often abbreviated to *process*); and
2. A *processing centre*.

As used within this study, a production process is defined as being a process involving the conversion of input resources to output resources; inputs must be consumed in order that outputs be produced. A processing centre is defined as being an aggregation of production processes which are similar in some way. The similarity is usually obviously apparent as Example 1.1 shows.

Example 1.1

Highly regulated forest stands (or crops) such as those found in commercial forest enterprises are usually even-aged and homogeneous in such other attributes as species composition, silvicultural tending, and harvesting method. The growth of such a stand over a number of time periods can be considered to be a production process (management alternative) where product formation (roundwood produced) is assumed to be made available by harvest of the stand during a time period. Such a process starts with afforestation or reforestation and ends with harvesting. For each process representing a specified stand, the process is characterised by a set of attributes that distinguish it from other processes. Among such attributes might be times to start and finish the process and the composition of wood resources obtained from the process. Each area available for management may have a number of stands proposed for it, each characterised by a different management process. Forests can be considered as aggregations of such areas. The spatial and temporal distribution of harvest volume is obtained by coordinating the harvest volumes of stands within the forest. Thus, a forest can be regarded as an aggregation of processes into a processing centre. //

The introductions of the notions of production processes and processing centres allows the resource allocation problems within an FUMP to be examined from the view of controlling activities within and between processing centres. Processing centres play a central role in "balancing" resources in FUMPs, in that

1. If resources are consumed at a processing centre in the formation of others, then consumption must be consistent

with resource availability.

2. If resources are transferred between processing centres, then resource conservation is ensured in the transfer.

As an example of point 1, the production of roundwood by processes within a centre must be done in a manner consistent with the availability of resources necessary for its production. These resources may include land, labour, and monies expended on roundwood production. This represents the balancing of resources within a processing centre.

This notion of balancing resources within and between processing centres shall be termed *integration*. Subsequently in this study, the expression *integration of processing centres*, or simply *integration* will be used.

The integration of processing centres has implications as to the possibility of decentralised planning at processing centres. This arises because the balancing, by processing centres, of shared and transferred resources implies that separate decisions cannot be made for each processing centre, then truly decentralised planning cannot exist. Thus, it is only by integration of processing centres that overall economic efficiencies can be pursued (Lasdon, 1970).

Processing centres can be integrated in various ways. These methods usually give consideration to quantities and/or prices of resources transferred. At the outset, then, the writer must emphasise that this study is concerned with methods of integration that are efficient in some well-defined economic sense. It is not concerned with methods of integration that involve concepts of equity unless these can be formulated explicitly and included into a model. The role of equity in the production and utilisation of forest resources is

complex. In New Zealand, the link between the size of a forest grower and the incorporation of equity notions into forest management is very strong.

The functioning of some mathematical programming algorithms that are used to identify optimal levels of economic efficiency for integrated processing centres can be explained in terms of successive specification of prices of shared resources until an optimality condition is satisfied (Dantzig and Wolfe, 1960). The most simple form of integration and the one most commonly practised is that of specifying requests or orders between processing centres. In this way, production and consumption levels for shared resources can be matched. As a result each processing centre is set a reasonably well-defined task as to what it must do.

As can be imagined, the number of feasible ways of obtaining integration between processing centres is immense. The question then arises, "Is it possible from among the many ways of integration to select better ways?" Selection may be achieved through making comparisons between alternative methods of integration and then making a better selection. In practice, this may be done by adopting a well-defined economic efficiency measure so that higher measures of efficiency are associated with better ways of integration.

Consideration will now be given as to why a mechanism that specifies integration as a requirement, and methodically searches for gains in economic efficiency is important in relation to FUMPs. The two extremes in the construction of planning models to represent the activities within a FUMP are simulation techniques and optimisation techniques. Generally simulation-based techniques are descriptive, in

that a scenario is specified and the model is run to set out in detail the activities of that scenario. Simulation techniques are usually computationally facile and cost little to run provided the detail demanded of them is not excessive. The limitations of these techniques, however, is that better scenarios may exist but may remain undetected in that they have not been explicitly formulated for examination. Alternatively, constrained optimisation techniques allow integration of processing centres through constraint specification, and the pursuit of economic efficiency through the generation of a sequence of feasible solutions with corresponding monotonic objective function values. The procedure ends when no method of integration can be found that is better than the incumbent. Conceptually, successful termination should always occur when the problem has been formulated correctly.

The possibility of advantage to be gained through application of optimisation techniques is what makes them appropriate vehicles to examine FUMPs. For this reason, the forest planning models developed and discussed in this study are largely optimisation based.

Mathematical programs representing FUMPs can become large (Clutter *et al.*, 1983), so the need for representations with facile resolution techniques is obvious. Because mathematical programs with linear constraint matrices are generally easy to solve, an initial decision to choose programs with linear constraint sets was made. (Note that this decision is consistent with the well-defined trend in applied mathematics to adopt approximation and linearisation in searching for solutions to problems). As will be discussed in chapter 3, certain FUMPs contain features that can be

resolved by the inclusion of integer variables into a model. For these problems, a Mixed Integer Linear Program (MILP) suffices for the purposes of representation (Murty, 1976).

To facilitate the examination of integration and efficiency within FUMPs, the study objectives in section 1.1 were outlined. These objectives directed the examination of FUMPs towards the related problem aspects of formulation, generation, solution and report writing. Section 1.2 contains a chapter summary and shows how this enquiry was pursued in succeeding chapters.

1.1 STUDY OBJECTIVES

The objectives of this study are set out in 1 through 4 below.

1. To formulate, using Linear Programming (LP) or Mixed Integer Linear Programming (MILP), a system characterising resource flows for a generalised Forest Utilisation Management Problem.
2. To identify means by which the classes of program representing Forest Utilisation Management Problems may be generated and solved.
3. To evaluate, for various formulations characterising components of Forest Utilisation Management Problems, features such as model generation, ease of solution, and information gained.
4. To examine critically the feasibility, the benefits, the difficulties and the drawbacks of using mathematical programming techniques to model Forest Utilisation Management problems.

The first of these objectives requires that the mechanisms that permit integration between and within processing centres be developed. Implicit in the statement of this objective is that the relationship between the integrative mechanisms and production models at processing centres be made clear.

The second objective leads to an examination of solution mechanisms in chapter 3 and the specification of a test problem in chapter 5.

The third objective leads to the identification in chapter 2, and the development in chapter 3 of various formulations of model components.

Finally, the fourth objective leads to a discussion in chapter 5 on the efficacy of using mathematical programming techniques for forest planning problems generally and in particular to the forest/utilisation planning problem.

1.2 OUTLINE OF CHAPTER CONTENTS IN THESIS

The remaining chapters in this study are described by content as follows.

In chapter 2, stand and forest level management mechanisms as means to govern resource flows (roundwood) from forests are examined. This is given in relation to linear programming representations of FMPs. Solution techniques for the LPs that arise are discussed.

Chapter 3 is concerned with the governing of resource flows that occur within the utilisation phase of a FUMP. Mechanisms to regulate resource flows at processing facilities (specialised processing centres) and markets are introduced and discussed. Solution techniques that give consideration

to structural aspects of FUMPs are also considered.

In chapter 4, methods that enable the possible prevention of redundancy in parts of an FUMP are examined (specifically, the prevention of redundancy in production models involving multistage processes is examined), the potential redundancy arising through resource unavailability. This has implications for matrix generation and may possibly allow the removal of integer variables from MILPs representing FUMPs.

Chapter 5 gives the discussion of a test problem used to test model components and procedures developed in chapters 2,3 and 4. In addition, general observations on matrix generators, report writers and forest planning models are made.

Chapter 6 contains the conclusions that can be drawn in relation to the study objectives proposed along with a discussion of related aspects of FUMPs.

CHAPTER 2

FOREST REGULATORY MECHANISMS

Solutions to the Forest Utilisation Management Problem (FUMP) require that the pattern of forest resources to be harvested over time be established and that the accompanying forest management activities be specified. In short, this involves forest level planning and the preparation of a harvest schedule.

Examined and extended in this chapter are some of the methods used in preparation of forest schedules. Collectively, these methods allow solution to a forest management problem. The approach initially is historical, concerned firstly with stand-level management techniques, then with the regulatory and structural aspects of forest-level planning mechanisms formulated as mathematical programs.

An outline of the sections is as follows: section 2.1 presents an historical outline of stand level management mechanisms; section 2.2 introduces the structural and regulatory aspects of forest-level management mechanisms; section 2.3 details a class of regulatory constraints that are both bounding and smoothing; section 2.4 discusses solution techniques employed on Forest Management Problems; and, section 2.5 discusses how regulatory constraints may be imposed in a FUMP.

2.1 STAND-LEVEL MANAGEMENT MECHANISMS

Stand-level management considers a forest as a number of stands, each of which is homogeneous for management purposes, usually age, species, and productive capacity. Each stand is then managed individually, using some criteria of efficiency, either economic or productive, to determine what constitutes the "best" stand management policy. The use of these productive and economic criteria will now be examined.

When stands are considered as homogeneous in age, species, and productive capacity and the question is to determine what rotation length will maximise the volume produced over an infinite time horizon, then one can easily show that, if successive rotations for the stand have the rotation age of maximum mean annual increment (MAI), then the volume production per unit area will be maximised (Clutter *et al.*, 1983). The use of any other rotation age will result in a lower average production rate (MAI) for each rotation, which will lower the total volume produced over an infinite time horizon. Appendix 2.1 documents some of the well known important relationships concerning mean annual increment (MAI) and current annual increment (CAI).

In the situation of determining which rotation age to use for a single crop type, Clutter *et al.*, (1983) cite the following decision criterion to maximise volume:

$$\max_t [\text{MAI}_t] = \max_t [Y_t/t] \quad (2.1)$$

where

t is the stand age
 Y_t is the volume per unit area at age t , and
 MAI_t is the mean annual increment at age t .

In choosing both crop-type and rotation age Clutter *et al.*, (*op.cit.*) use the criterion.

$$\max_k \left[\max_{t_k} [MAI_{t_k}] \right] = \max_k \left[\max_{t_k} [Y_{t_k} / t_k] \right] \quad (2.2)$$

where

t_k is the stand age of crop type k
 Y_{t_k} is the volume per unit at age t for crop type k, and
 MAI_{t_k} is the mean annual increment at time t for crop type k.

These decision criteria assume an infinite planning horizon and also assume that any crop type, once initially established, will be reafforested after each harvest. Both these assumptions are usually relaxed when considering forest level management techniques, in that both a finite time horizon and transferral of areas between crop-types are permissible.

Volumetric decision criteria such as (2.1) and (2.2) seldom suffice as adequate criteria for selecting stand-level management methods, in that most management objectives are ultimately economic in nature. Instead, economic criteria that account for the time value of money and the flow of costs and returns that may occur over a rotation or planning horizon are more attractive as decision criteria. Those criteria that consider the flow of costs and returns over a rotation all require as input a statement of costs and returns for the proposed rotation as in Table 2.1.

Those methods using a planning period require a statement of costs and revenues for each year of the planning period. Given such information, the most common economic

decision criteria utilise as measures a discounted sequence of net revenues, discounting being adopted because it incorporates the time value of money and allows comparison of projects that may terminate at different points in time.

Table 2.1 : Tableau of Crop-Type Revenues and Costs

Tabulated for a specified crop-type are the revenue and cost flows during a rotation of length n years.

t	revenue @ t	cost @ t
0	r_0	c_0
1	r_1	c_1
2	r_2	c_2
.	.	.
.	.	.
.	.	.
n	r_n	c_n

where

r_i is the revenue per unit area in year i

c_i is the cost per unit area in year i .

One common rotation-based economic measure is land expectation value (LEV), which considers the present value of an infinite sequence of rotations, each having the same cost and revenue structure.

$$LEV_t = \frac{\sum_{j=0}^t (r_j - c_j) (1+i)^{t-j}}{(1+i)^t - 1} \quad (2.3)$$

where

- LEV_t is the land expectation value in monetary units for a rotation length of time t , (this is also known as soil expectation value);
- r_j is the revenue associated with time j of the rotation;
- c_j is the cost associated with time j of the rotation;
- t is the rotation length in time units¹; and
- i is the discount rate per unit time.

The following measures can either be rotation or planning-horizon based, they are defined by (2.4) and (2.5) to be the sum of the sequence of discounted or compounded net revenues respectively.

$$NPV = \sum_{j=0}^t \frac{(r_j - c_j)}{(1+i)^j} \quad (2.4)$$

$$NFV = \sum_{j=0}^t \frac{(r_j - c_j)}{(1+i)^{j-t}} \quad (2.5)$$

where

- NPV is the net present value in monetary units,
- NFV is the net future value in monetary units,
- r_j is the revenue per unit area in time j , and
- c_j is the cost per unit area in time j .

In the general case of choosing both crop-type and rotation age, the following economic decision criterion can be formulated given measure (2.3)

¹ A distinction is sometimes made between rotation age and rotation length is always greater than or equal to rotation age because it is taken to include the activities of site preparation and planting.

$$\max_k \left[\max_{t_k} [LEV_{t_k}] \right] \quad (2.6)$$

However, decision criteria similar to the measures (2.4) and (2.5) must be applied with care in that they include a fraction of a rotation if the measures are applied to a planning horizon that is not an integral number of rotation lengths.

The use of an economic decision measure implies that a ranking can be stated as to the relative desirability of proposed rotations. However, in general, the rankings derived from employing differing economic measures will be different. Equations (2.3) and (2.4) each suggest a ranking relation, with ties being broken arbitrarily. These relations are as follows:

if $LEV_{t_k} > LEV_{q_r}$
 then choose crop-type k
 with rotation length t;

if $NPV_{t_k} > NPV_{q_r}$
 then choose crop-type k
 with rotation length t.

Each of these ranking relations allows the construction of the preferred ordering of the rotations proposed for a stand. However, the preferred orderings, or rankings, are not necessarily the same when different ranking relations are used. Given the above ranking relations, the following is possible:

(see Table 8.2, p217 Clutter *et al.*, *op. cit.*)

$$LEV_{t_k} > LEV_{q_r}$$

and

$$NPV_{t_k} < NPV_{q_r} .$$

Thus, the sets of available economic measures are not necessarily consistent in their rankings, and debate surrounds the choice of measures. Forest level management is obtained by the integration of stand-level management. The inability to generate consistent rankings under all measures carries over to the forest-level management problem, and also to the forest utilisation management problem.

2.2 FOREST LEVEL MANAGEMENT MECHANISMS

Concepts from stand level planning techniques play an important part in forest-level planning techniques because the latter can be considered to involve joint management of the former. The joint management or integration being required in order to satisfy production smoothing requirements. The smoothing considerations mean that decisions can no longer be made independently for each stand and that decision making techniques that are capable of integrating the production from stands so as to meet the smoothing requirements must also be used.

Forest-level planning should identify the sequence of actions to take place over the planning horizon that are consistent with specified smoothing requirements and should try to meet other objectives that management may specify. The outcome of this planning is a document called a *cutting schedule*, which prescribes both the actions to be taken and their timing over the planning horizon. Historically,

target forest concepts played an important role in the preparation of cutting schedules. For example, a "normal forest" was considered to be an idealised structure whose attainment and subsequent maintenance was considered to constitute good management practice. Such considerations are, at the time of writing, regarded as anachronistic. However, target forest concepts still remain in forest planning, with "fully regulated forests" (cited in Clutter *et al.*, 1983) and "equivalent normal forests" (Allison *et al.*, 1979) having arisen to describe what are considered suitable target situations. Japanese forest planners still give strong consideration to target forest concepts (Suzuki, 1984; Choi and Nagumo, 1984), and mathematical programs designed to attain a specified target forest structure from a given initial forest structure have been formulated (Choi and Nagumo, *op. cit.*).

Most forest-level management techniques currently allow what Clutter *et al.* (1983) term "the intelligent management of imbalanced forest structures". Loosely speaking, these can be considered to be a relaxation of target forest concepts, where the allowed targets are not so rigorously defined. The motivation for the relaxation is that stable forest communities, other than those based on target forest concepts, can exist and moreover these communities may be attractive economically.

This section details mechanisms used in forest-level management that may be formulated using mathematical programming techniques. Subsection 2.2.1 introduces the important concept of a management unit and its possible representations; subsection 2.2.2 presents the structural constraints necessary for different management unit

representations; subsection 2.2.3 introduces the regulatory constraints to achieve integration of management units; and, regulatory constraints necessary to deal with multiple crop-types.

2.2.1 Representation of Management Units

Assuming forest level management is concerned with management of stands, each of which is homogeneous with respect to crop-type, age class, and productive capacity, then forests with such stands can be partitioned into crop-type age classes because each such stand can be allocated to only one crop-type age class. Different stands belong to different crop-type age classes.

The partition of an area into its constituent crop-type age classes can change over time because forest growth is characterised by the transfer of areas between age classes and possibly crop-types over time as areas are harvested and reafforested. The current definitions for a management unit², model I and model II (Johnson and Scheurman, 1977) are defined in relation to the area partitions induced by crop-type age classes. They differ in that model II formulations consider how this partition may change over time.

A model I formulation does not use area transfers between crop-types as part of its definition. Instead, it uses the crop-type age class area partition induced in the initial planning (model) period to define management units. That is, each crop-type age class having a non-zero area in the initial planning period constitutes a management unit. A model II formulation, on the other hand uses the area partition induced in each planning period to define

² Ware and Clutter (1971) term such devices cutting units. In this study, they shall be termed management units, which implies that other management activities besides harvesting may take place.

management units. Thus implicitly incorporating the area transferred³ between age classes and crop-types as part of the definition.

In a model I formulation, strategies are proposed for a management unit, whereas for a model II formulation, the strategies proposed determine the management units that may arise during the planning horizon. Model I management units always have an area associated with them, whereas for those model II units formed during the planning horizons, this is not necessarily so.

Figure 2.1 shows the management alternatives proposed for a single model I management unit. These same alternatives would represent seven management units in a model II formulation; the initial management unit established at least two periods before period 1, and the six management units arising from harvest of the initial crop-type in periods 1 through 6, or from harvest of a subsequent crop-type in periods 3 through 6.

2.2.2 Single Crop-Type Structural Constraints

The constraints used to express conservation of area for management units that arise in a model I or model II formulation are termed structural constraints, whereas those constraints that integrate the flow of resources from management units shall be termed regulatory. It is not surprising that differing sets of structural constraints arise for Model I

³ Crop-type age class transfers between periods can occur as follows:

- No harvest occurs and the subsequent age class is entered the subsequent period;
- A harvest occurs with reforestation of the same crop-type, and the first age class is entered in the subsequent period;
- A harvest occurs with reforestation of a different crop-type, and the first age class is entered in the subsequent period.

Management alternative	Period Number					
	1	2	3	4	5	6
1	H		H		H	
2	H		H			H
3	H		H			
4	H			H		H
5	H			H		
6	H				H	
7	H					H
8	H					
9		H		H		H
10		H		H		
11		H			H	
12		H				H
13		H				
14			H		H	
15			H			H
16			H			
17				H		
18				H		
19					H	
20						H

Figure 2.1 : Alternatives for a model I management unit.

The management unit arises from a crop-type age class containing harvestable volume in the initial planning period. The minimum rotation length is two periods, H denotes the harvest/reaforestation of the same crop-type.

(Source: Johnson and Scheurman, 1977).

and model II management units since the units are defined in different ways in terms of the area partitions induced by the crop-type age classes.

In the case of a single crop-type, Johnson and Scheurman (1977) define the model I and model II structured constraints respectively as equations (2.7) and (2.8)

$$\sum_{q=1}^{R_\ell} x_{\ell q} = A_\ell \quad \forall \ell \quad (2.7)$$

$$\ell = 1, \dots, U$$

where

$x_{\ell q}$ are the units of area of management unit ℓ assigned management alternative q ,

A_ℓ are the units of area of management unit ℓ ,

U is the number of management units, and

R_ℓ is the number of management alternatives for management unit ℓ .

$$\begin{aligned} \sum_{j=1}^N x_{ij} + w_{iN} &= A_i \quad \forall i \\ \sum_{k=j+z}^N x_{jk} + w_{jN} &= \sum_{i=-M}^{j-z} x_{ij} \quad \forall j \end{aligned} \quad (2.8)$$

$$i = -M, \dots, 0$$

$$j = 1, \dots, N$$

where

x_{ij} (x_{jk}) are the units of area reforested in period i (period j) followed by harvest/reforestation in period j (period k);

w_{iN} (w_{jN}) are the units of area reforested in period i (period j) and left as part of the ending inventory in period N ;

A_i are the units of area present in period one that were either afforested or reforested in period i , with each A_i being a constant at the beginning of the planning horizon (period 1);

N is the number of periods in the planning horizon;

M is the number of periods before period zero in which the oldest age class in period one was afforested or reforested;

z is the minimum number of periods between harvest reforestation.

2.2.3 Single Crop Type Regulatory Constraints

Constraints to integrate the flow of resources from management units, the so called regulatory constraints, may include harvest area; harvest volume; residual area; residual volume; and area transfer constraints. These can all be formulated as linear combinations of the activities in a model I or model II formulation, and they by no means exhaust the set of meaningful combinations that can be formed for regulatory purposes (Garcia, 1984). Historically, foresters have been concerned with harvest volume or harvest area constraints as the chief regulatory mechanisms. Johnson and Scheurman (1977) present (2.9) as harvest volume constraints, which are smoothing but not bounding, in that they smooth the flow of harvest volume between periods but do not bound it explicitly at each period.

Model I harvest volume expression

$$h_j = \sum_{\ell=1}^U \sum_{q=1}^{R_\ell} V_{\ell q j} x_{\ell q}$$

where

h_j is the total harvest volume in period j ,

$V_{\ell q j}$ is the volume per unit area harvested in period j , from management unit ℓ under management alternative q .

Model II harvest volume expression

$$h_j = \sum_{k=j}^N \sum_{i=-M}^{j-2} V_{ijk} x_{ik} + \sum_{i=-M}^N V_{ijN} w_{iN}$$

where

h_j is the total harvest volume in period j ; and

V_{ijk} is the volume per unit area arising in period j with afforestation or reafforestation in period i with clear felling in period k .⁴

⁴ This definition permits the inclusion of intermediate harvests such as extraction thinnings.

Given expressions for harvest volume, the following constitutes a set of harvest volume smoothing constraints.

$$\begin{aligned}
 (1-\alpha)h_j - h_{j+1} &\leq 0 & \forall j \\
 (1+\beta)h_j - h_{j+1} &\geq 0 & \forall j \\
 j &= 1, \dots, N-1
 \end{aligned} \tag{2.9}$$

where

α is the maximum decrease in harvest volume from period to period.
 (For example, $\alpha = 0.10$ implies a maximum decrease of 10 per cent).
 β is the maximum increase in harvest volume from period to period.

Section 2.3 details methods by which constraint sets such as (2.9) can be made both smoothing and bounding, when the resource being regulated is assumed to be a generalised linear combination of the activities (structural variables) in a model I or model II formulation.

2.2.4 Multiple Crop Type Structural Constraints

The structural constraints given in subsection 2.2.2 assume the existence of a single crop-type. The extension of models I and II for multiple crop-types is given in this section⁵.

Model I multiple crop-types structural constraints

$$\sum_{k=1}^{K_{ij}} \sum_{\ell=1}^{L_{ijk}} x_{ijk\ell} = A_{ij} \quad \forall i, j \tag{2.10}$$

$$i = 1, \dots, I$$

$$j = 1, \dots, J_i$$

where

$x_{ijk\ell}$ are the units of area for the management unit defined in the initial period by crop-type i , age class j , managed under strategy ℓ that uses crop-type k for reafforestation;

A_{ij} are the units of area for the management unit defined in the initial period by crop-type i , age class j .

⁵ The identifiers and subscripts used in this section depart in meaning from their previous usage, but the meaning of the new usage is made apparent.

K_{ij} is the crop-type reafforestation set for the measurement unit defined by crop-type i and age class j ; and

L_{ijk} is the management alternative set.

The structural constraints (2.10) permit transfers between crop-types within a management alternative. The number of crop-type transfers per management alternative is restricted to be at most one. If present, this transfer will occur upon harvest of the crop-type that is resident in the first planning period and involves reafforestation with an alternate crop-type. This can be envisaged by extending Figure 2.1 as follows. The first harvest in strategies 1 through 20 is of the initial crop-type and all subsequent re-afforestation in these strategies uses a different crop-type. The decision to restrict the number of crop transferrals to at most one per management alternative was made in order to reduce the number of alternatives (variables) required by this formulation. Figure 2.2 is digraph representing the structural constraints (2.10) when more than one crop-type is present.

Any model I management unit can be considered to be a network as in Figure 2.2. Generally, the harvest patterns will differ depending on the establishment period, the minimum and maximum ages of clearfelling for the initial and subsequent crop-types. Because paths in the network have to be constructed for each of the subsequent crop-types, the number of variables required to formulate a model I approach quickly increases.

A structural constraint set to extend model II to deal with multiple crop-types has been proposed by Garcia (1984). These are equations (2.11) through (2.14). Figure 2.3 is

Management alternative	Period Number					
	1	2	3	4	5	6

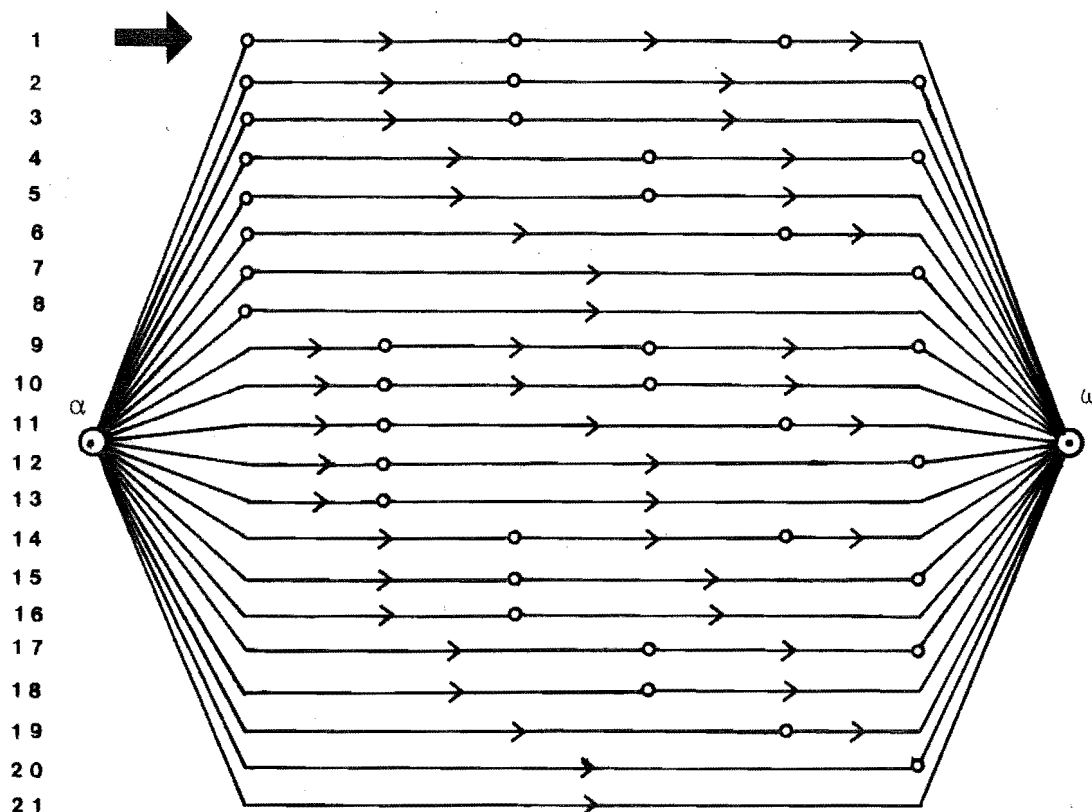


Figure 2.2 : Diagraph showing model I management alternatives

The harvest pattern for the alternatives 1 through 21 in Figure 2.1 is indicated. Vertices (when present) along each path from the vertex labelled α to the vertex labelled ω denote a harvest. Paths between α and the first harvest vertex (when present) correspond to the growth of the initial crop-type, subsequently each path represents growth and possible harvest of the reafforested crop-type.

an example of a digraph corresponding to such a constraint set.

Model II multiple crop-types structural constraints

$$\sum_j Y_{tij} = \sum_k r_{tik} \quad \forall i, t \quad (2.11)$$

$$\sum_k r_{tki} = \sum_{s=t+1}^{T+1} Y_{s,i,s-t} \quad \forall i, t \quad (2.12)$$

$$a_{ij} + \sum_k z_{kji} - \sum_k z_{ijk} = \sum_{s=1}^{T+1} Y_{s,i,j+s-1} \quad \forall i, j \quad (2.13)$$

$$Y_{tij}, r_{tik}, z_{ijk} \geq 0 \quad (2.14)$$

$$i = 1, \dots, I; \quad j = 1, \dots, J; \quad t = 1, \dots, T$$

where

Y_{tij} are the units of area clear felled in period t from crop-type i and age class j ;

r_{tik} are the units of area clear felled in period t from crop-type i and immediately replanted into crop-type k ;

z_{ijk} are the units of area transferred from crop-type i and age class j to crop-type k . These transfers are made at the commencement of the initial planning period;

a_{ij} is the initial area available of crop type i and age class j ;

I is the number of crop-types either initial or reafforested;

J is the oldest initial age class; and

T is the number of planning periods.

Equation (2.11) implies that, for crop-type i in any period t , the subscript j runs up to the oldest initial age class J . In general, for period t , the subscript j must range over the age classes that may be present in period t and may be harvested. Equation (2.11), in terms of the oldest initial age class, should then be specified as

$$\sum_{j=1}^{J+t-1} Y_{tij} = \sum_k r_{tik} \quad \forall i, t \quad (2.15)$$

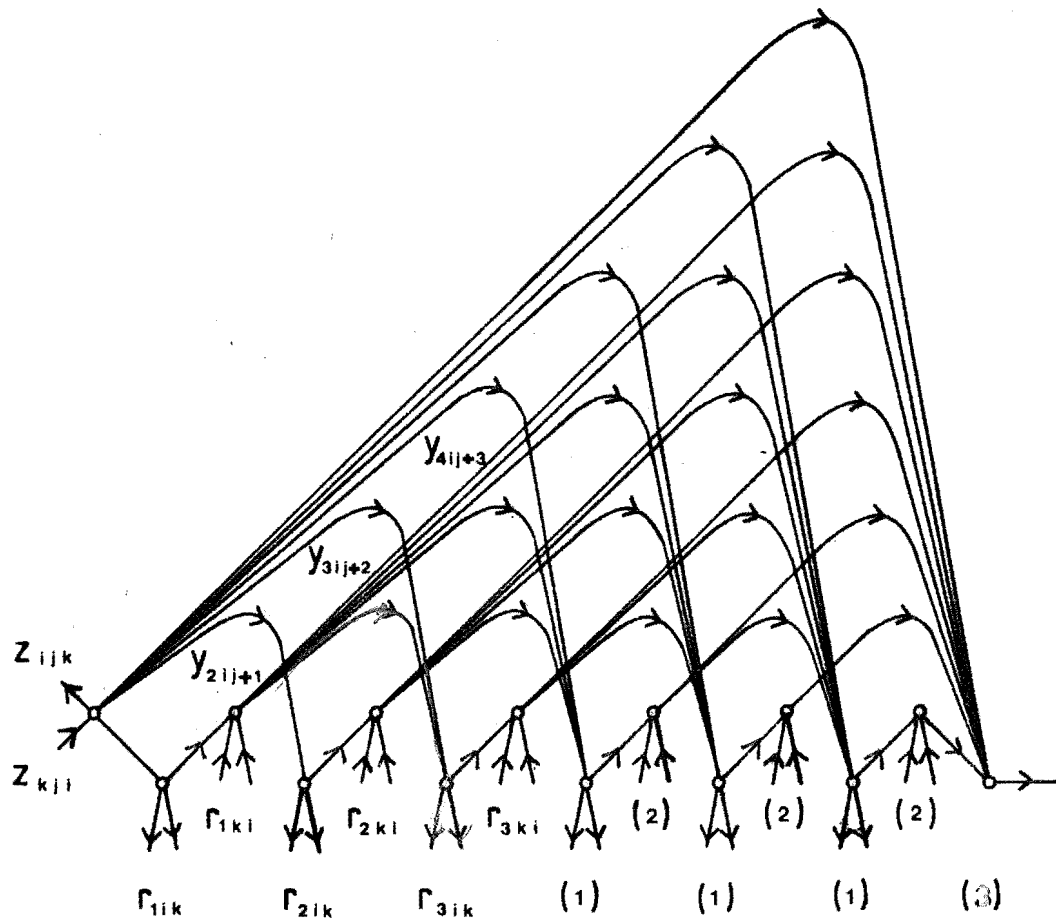


Figure 2.3 : Digraph showing model II management alternatives

A single crop-type i , initial age class j , is scheduled over six periods using equations (2.12) through (2.15). The crop-type is of harvest age in the initial planning period. Subsequent rotations of the same crop-type have a minimum rotation length of two periods (c.f., Figure 2.1).

The digraph in Figure 2.3 represents exactly the same management possibilities as found in Figure 2.1. Equations (2.15), (2.12), and (2.13) represent the areas entering and leaving vertices of the types labelled (1), (2), and (3) respectively in Figure 2.3. The dangling arcs, that is, the arcs without vertices attached, are included for completeness only, (r_{tik} 's, z_{ijk} 's) in that they would all have value zero in this example. Arcs of the form r_{tii} are not dangling and may or may not have value zero, depending on whether harvest of crop-type i has occurred in period t .

By inspection of Figures 2.2 and 2.3 it is evident that they do not have the same structure [they are not isomorphic to each other, (see Robinson and Foulds, 1980)], in that each vertex in Figure 2.2, excepting the source and the sink, has one arc entering and one arc leaving, whereas this does not hold in Figure 2.3. Clearly, digraphs such as these can be used to show the difference in representing model I and model II management units.

The variables actually specified in the set of structural constraints depend on the age classes initially present for each crop-type, the minimum and maximum age of clearfelling, and area transfers. This will always be less than or equal to the number of variables specified by equations (2.12) through (2.15). Thus, these equations specify a superset of the variables actually used. The structural constraints are presumably specified in this way in order to reduce the subscripting complexity that would arise should an attempt be made to subscript the variables actually used. Garcia (*op. cit.*) uses a computer program that specifies as part of its input the necessary subscripting information to generate the required problem and, in this way, overcomes the difficulty of working with a perhaps difficult set of subscripts.

The model I and model II formulations differ in terms of the number of variables required to represent them. As the size of the problem becomes larger, the model II formulation becomes more efficient in terms of the number of variables used. The following procedures can be used to enumerate the variables required by each model.

The number of variables required by a model I formulation is given by summing over all management units the

number of paths in the digraph representing the structural constraints. On the other hand, for a model II formulation, it is equal to the number of arcs in the digraph representing equations (2.12) through (2.15). The problem of counting these arcs can be simplified. If these equations were to consider only a single crop-type then all r_{tik} 's and z_{ijk} 's could be dropped. Thus, in Figure 4, the vertices labelled (1) and (2) would become the same (co-incident) and the number of arcs in the remaining subdigraph could be computed using the formulae of Johnson and Scheurman (1977). Clearly, the number of variables needed for a model II with multiple crop-types could be determined by using the formulae to calculate the number of y_{tij} 's for each crop-type, aggregating over crop-types, and including the number of ways of transferring area between crop-types, that is, the number of r_{tik} 's and z_{ijk} 's.

2.2.5 Multiple Crop-type Regulatory Constraints

As indicated in subsection 2.3.3, linear combinations of the structural variables within a model I and II formulation that can be used for regulatory purposes are easy to construct. To facilitate the specification of these for model II, Garcia (1984) specifies the residual area after the harvest of crop-type i age class j in period t to be given by equation (2.16)

$$x_{tij} = \sum_{s=t+1}^{T+1} Y_{s,i,j+s-t} \quad (2.16)$$

where

x_{tij} is the residual area after harvest of crop-type i age class j in period t .

The linear combinations for total harvest volumes for model I and model II are given respectively by equations (2.17) and (2.18)

$$h_n = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{\ell=1}^L V_{ijk\ell n} X_{ijk\ell} \quad (2.17)$$

$$h_n = \sum_{i=1}^I \sum_{j=1}^{J+n-1} \sum_{t=n}^{T+1} V_{tijn} Y_{tij} \quad (2.18)^6$$

where

$V_{ijk\ell n}$ is the volume per unit area produced in period n by management alternative ℓ , whose initial crop-type is i , initial age class j , and whose regeneration crop-type is k

V_{tijn} is the volume per unit area produced in period n by crop-type i that is clearfelled in period t when in age class j .

The implementation of regulatory constraints usually requires the concise definition of the resource sets that are being regulated. Typically a resource can be any commodity that characterises forest growth and is readily quantifiable. This criterion of quantifiability is particularly important in that the inclusion of imprecise data items in mathematical programs representing forest management problems can have deleterious effects (Rose, 1984). This suggests that suitable resources might be those commodities that can be measured in terms of area or volume, these being the most precise measures available for characterising forest growth. Consider, for example, a log class set that is rigidly defined by log grading rules, then the volumes either standing or harvested for members of the log class set could

⁶ Not all terms in the summation are necessarily defined. The index set in (2.18) may be a superset of the index set actually used.

be used as a resource set. Similarly, the areas of crop-types either standing or harvested in any period can be used as a resource set.

The choice of what constitutes a suitable resource set to describe output from a forest is, in New Zealand, a matter of contention at the time of writing. However, within models representing forest management problems or forest utilisation management problems, the definition of such resources sets is crucial. It governs the description of material to be received by processing centres that utilise forest resources and, for FUMPs, the definition of processes within those centres. A discussion of the attributes suitable for the classification of resources output from a forest, in the case when resources are considered to be logs, is given in section 3.1 of Chapter 3. These attributes must be sufficiently detailed so as the various log input requirements of different utilisation processing centres can be specified in terms of the attributes.

2.3 BOUNDING AND SMOOTHING REGULATORY CONSTRAINTS

Provided that the resources to be regulated are adequately defined and can be expressed as linear combinations of the structural variables within a model I or model II formulation, then regulatory constraints can be imposed. Generally, such constraints are considered to be either smoothing or bounding. This section considers the development of a class of constraint that is both smoothing and bounding. The development proceeds via a statement of a generalised smoothing constraint set, an examination of its properties, and a statement of the extensions necessary for the bounding action.

Smoothing constraint set

$$(1+\gamma_{ij})r_{ij} - r_{ij+1} \leq 0 \quad \forall i,j \quad (2.19)$$

$$(1+\delta_{ij})r_{ij} - r_{ij+1} \geq 0 \quad \forall i,j \quad (2.20)$$

i is the index of the i^{th} resource

$j=1, \dots, N-1$, is the index from the time set

where

γ_{ij}, δ_{ij} are scalars such that $(1+\gamma_{ij}), (1+\delta_{ij}) > 0$, and r_{ij} is the measure of resource i at time j , and $r_{ij} \geq 0$

Consider the following in relation to (2.19) above⁷

Case one

$r_k = 0$ for $k \in \{2, \dots, N\}$
 if $r_k = 0$ then by (2.19)
 $r_j = 0$ for $1 \leq j \leq k-1$ since $r_j \geq 0, (1+\gamma_j) \geq 0$.

Case two

$r_k > 0$ for $k \in \{1, \dots, N-1\}$
 if $r_k > 0$ then by (2.19)
 $r_j > 0$ for $k+1 \leq j \leq N$.

Case one suggests that, should the resource measure at any period, except the first, become zero, then the resource at all previous periods is constrained to be zero. Case two suggests that, should the resource measure at any period, except the last, become positive, then the resource measure at all succeeding periods will become positive.

Bounding constraints can be formulated using the system (2.19), which implicitly satisfy the systems (2.21) and (2.25). The proofs are given in the following lemmas.

⁷ The subscript i is dropped. This may be done without any loss of generality in that what follows applies to any resource i .

Lemma 2.1

Given the constraint system (2.19), show that

$$r_{j+1} \geq (1+\gamma_1)(1+\gamma_2)\dots(1+\gamma_j)r_1 \quad (2.21)$$

for $j=1,\dots,N-1$.

Proof: by induction

$P(1)$ is obviously true from the system (2.19), assume $P(k)$ true, $k = 1,\dots,N-2$. An induction proof requires that $P(k+1)$ holds. $P(k)$ true implies (2.22)

$$r_{k+1} \geq (1+\gamma_1)(1+\gamma_2)\dots(1+\gamma_k)r_1 \quad (2.22)$$

multiply both sides of (2.22) by $(1+\gamma_{k+1})$, then

$$(1+\gamma_{k+1})r_{k+1} \geq (1+\gamma_1)(1+\gamma_2)\dots(1+\gamma_{k+1})r_1 \quad (2.23)$$

Consider the constraint system with $j=k+1$, then

$$r_{k+2} \geq (1+\gamma_{k+1})r_{k+1} \quad (2.24)$$

Jointly, (2.23) and (2.24) imply $P(k+1)$. \square

Lemma 2.2

Given the constraint system (2.19), show that

$$r_{N-j} \leq \frac{r_N}{(1+\gamma_{N-j})\dots(1+\gamma_{N-2})(1+\gamma_{N-1})} \quad (2.25)$$

$j = 1,\dots,N-1$.

Proof: by induction

To establish $P(1)$ set $j = N-1$ in the constraint system, then

$$r_{N-1} \leq \frac{r_N}{1+\gamma_{N-1}}$$

and $P(1)$ is true, assume $P(k)$ true for $k=1,\dots,n-2$, $P(k)$ true implies

$$r_{N-k} \leq \frac{r_N}{(1+\gamma_{N-k})\dots(1+\gamma_{N-2})(1+\gamma_{N-1})} \quad (2.26)$$

Multiply both sides of (2.26) by $\frac{1}{(1+\gamma_{N-(k+1)})}$, then

$$\frac{r_{N-k}}{(1+\gamma_{N-(k+1)})} \leq \frac{r_N}{(1+\gamma_{N-(k+1)}) \dots (1+\gamma_{N-2})(1+\gamma_{N-1})} \quad (2.27)$$

Consider the system (2.19) with $j = k+1$, then

$$r_{N-(k+1)} \leq \frac{r_{N-k}}{(1+\gamma_{N-(k+1)})} \quad (2.28)$$

Jointly, (2.27) and (2.28) imply $P(k+1)$ □

Lemma 2.1 suggests the system (2.19) will be bounding and smoothing provided an additional constraint of the form (2.29) is added. Similarly, Lemma 2.2 suggests that, provided a constraint of the form (2.30) is added, the system will be bounding and smoothing⁸, an example is shown in Figure 2.4.

$$r_1 \geq \psi^{\min} \quad (2.29)$$

$$r_N \leq \psi^{\max} \quad (2.30)$$

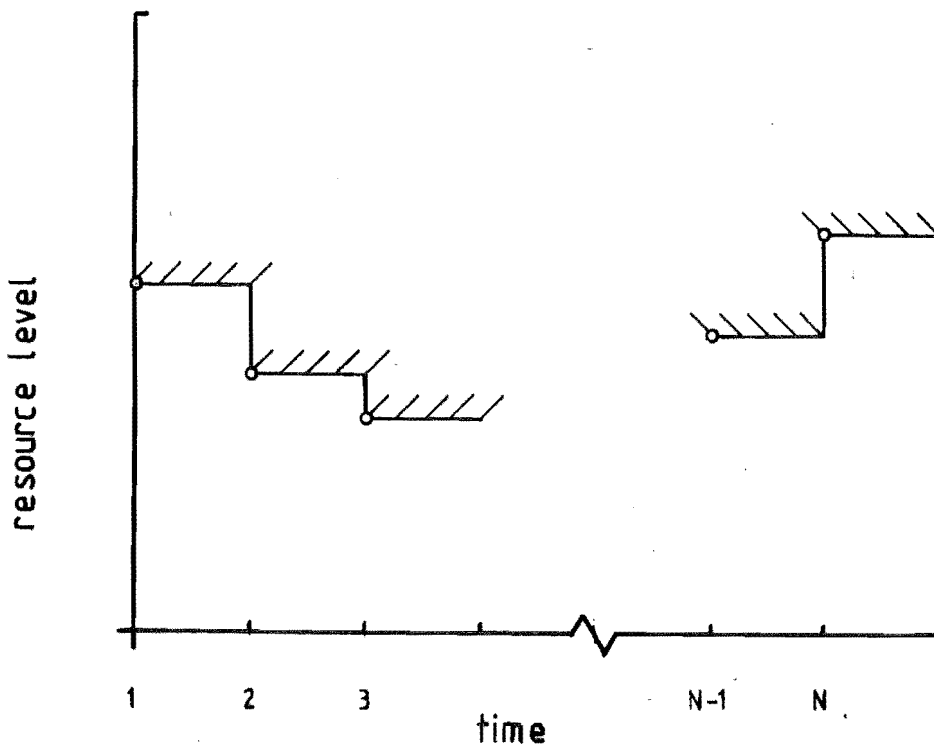


Figure 2.4 : A Resource Lower Bound Envelope

The envelope is formed using (2.19) and (2.29), feasible resource amounts at each model period lie above the shaded horizontal lines.

⁸ It is an easy matter to implement (2.29) or (2.30) as bounded variables within the simplex procedure.

Similarly, the system (2.20) can be used to develop constraints that are both bounding and smoothing, however, the behaviour of this system is initially examined in cases three and four below.

Case three

$$\begin{aligned} r_k &= 0 \quad \text{for } k = 1, \dots, N-1 \\ \text{if } r_k &= 0 \quad \text{then by (2.20) and induction} \\ r_j &= 0 \quad \text{for } k+1 \leq j \leq N. \end{aligned}$$

Case four

$$\begin{aligned} r_k &> 0 \quad \text{for } k = 2, \dots, N \\ \text{if } r_k &> 0 \quad \text{then by (2.20) and induction} \\ r_j &> 0 \quad \text{for } 1 \leq j \leq k-1. \end{aligned}$$

Case three suggests that, should the resource measure at any period, except the last, become zero, then the resource measure at all succeeding periods will be zero. While Case four suggests, that should the resource measure at any period, except the first, become positive, then the resource measure in all previous periods is constrained to be positive.

The following Lemmas arise in conjunction with (2.20).

Lemma 2.3

Given the constraint system (2.20), then

$$\begin{aligned} r_{j+1} &\leq (1+\delta_1)(1+\delta_2)\dots(1+\delta_j)r_1 \quad \forall j \\ j &= 1, \dots, N-1. \end{aligned} \tag{2.31}$$

Proof (omitted)

The proof is by induction and closely parallels the proof of Lemma 2.1 □

Lemma 2.4

Given the constraint system (2.20), then

$$\begin{aligned} r_{N-j} &\geq \frac{r_N}{(1+\delta_{N-j})\dots(1+\delta_{N-2})(1+\delta_{N-1})} \quad \forall j \\ j &= 1, \dots, N-1. \end{aligned} \tag{2.32}$$

Proof (omitted)

The proof is by induction and closely parallels the proof of Lemma 2.2 □

Lemmas 2.3 and 2.4 suggest that the system (2.20) will be both smoothing and bounding, provided either or both of the constraints (2.33) and (2.34) are added. An example is shown in Figure 2.5.

$$r_1 \leq \phi^{\max} \quad (2.33)$$

$$r_N \geq \phi^{\min} \quad (2.34)$$

where

ϕ^{\max} is an upper bound for the resource in the first period, and
 ϕ^{\min} is a lower bound for the resource in the last period.

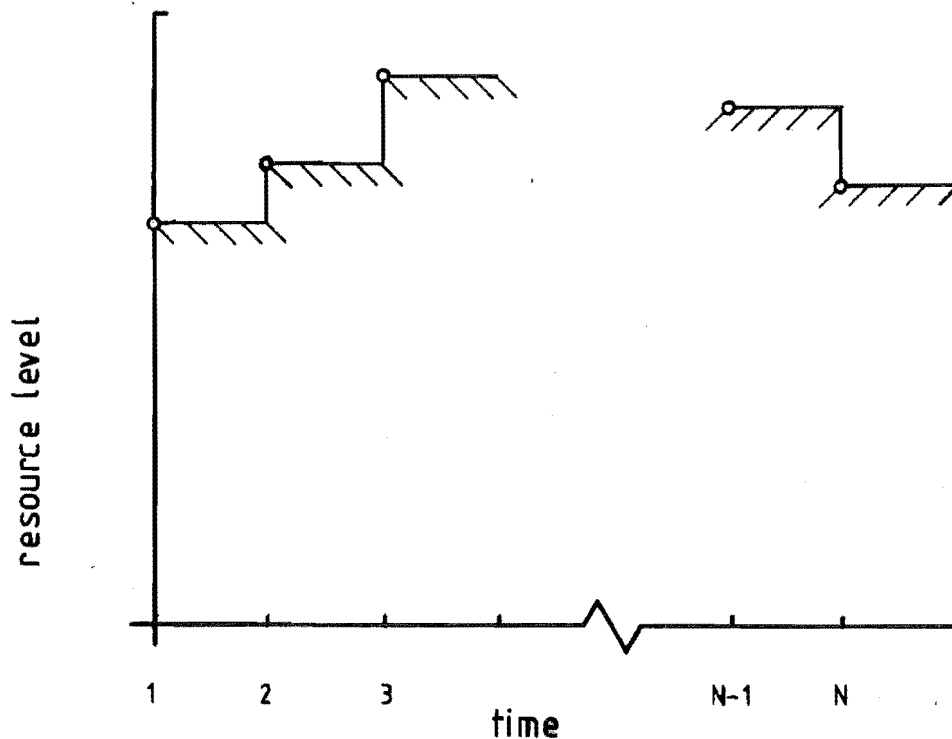


Figure 2.5 : A Resource Upper Bound Envelope

The envelope is formed using (2.20) and (2.33), feasible resource amounts of each model period lie below the shaded horizontal lines.

In applying systems of constraints that are both bounding and smoothing, a reduction of constraints may be achieved from the situation where the bounding and smoothing constraints are applied separately. However,

application requires that careful consideration be given to how the bounding and smoothing system operates. This may be done by considering cases one through four above. The choice of which system to apply might take into account such aspects as the accuracy of the bound. For example, it may be easier to determine a bound for the initial period because the estimation period is shorter and the precision of estimate is likely to be higher than for the final period. Consideration could also be given to applying such constraints over a subset of time periods, creating partial envelopes of smoothing and bounding constraints. Finally, it may be better to only smooth some resources rather than smooth and bound them. For example, depending on how the constraints are formulated the addition of bounding constraints may unnecessarily impose a pattern of resource consumption or production, in all model periods.

2.4 SOLUTION TECHNIQUES FOR FOREST MANAGEMENT PROBLEMS

Historically, the development of mathematical programs to represent forest management problems have evolved from a class of programs that allowed only a narrow range of management criteria to be specified to one that permits both generalised structural and regulatory constraints and is typified by the models of Hoganson and Rose (1984) and Garcia (1984) which are respectively model I and model II in terms of their structural constraints.

Similarly, the methods used to solve FMPs have evolved from basic application of the simplex method (Leak, 1964; Wardle, 1965; Navon and McConnen, 1967; Ware and Clutter, 1971) to methods that use the structural aspects of the FMP as the basis for decomposition techniques (Tcheng, 1966;

Berck and Bible, 1984; Garcia, 1984). A method that considers both structural aspects and the possible imprecision and inaccuracy of data items has also been proposed (Hoganson and Rose, 1984).

This section discusses solution techniques, their evolution in relation to the class of FMP being solved, and the method employed. Subsection 2.4.1 discusses applications of the simplex method; subsection 2.4.2 discusses a model I decomposition technique; subsection 2.4.3 discusses a model II decomposition technique; and subsection 2.4.4 discusses a solution method based on the estimation of certain key dual variables.

2.4.1 Standard Simplex Methods

Conceptually, all linear programming problems can be solved by application of the simplex procedure, but practical limits exist as to the size of problem that can be solved using such central algorithms as the revised simplex method. At the time of writing, the magnitudes of 20,000 variables and 15,000 constraints (approximately) define the range of the revised simplex procedure⁹.

Currently, major linear programming generators and report writers are used for generating and reporting on FMPs that are to be solved using the revised simplex procedure. Amongst these are the systems, MAX-MILLION (Clutter, 1968), TIMBER RAM (Navon, 1971) and FORPLAN (Johnson *et al.*, 1980). Although the development work for such systems was chiefly undertaken in the United States, work has also been undertaken in Australia (Paine, 1966, cited in Clutter *et al.*, 1983) and New Zealand (Shirley, 1978; Garcia, 1984).

9

Compact inverse techniques and sparse matrix techniques are sometimes used in conjunction with the revised simplex method to increase its efficiency. However, the problem of size still remains.

Johnson and Scheurman (1977) consider solution techniques to various simple FMPs, that would otherwise be solved by simplex methods, by specification of the Kuhn Tucker conditions (Walsh, 1975). Where the objective function is linear or quadratic and the constraint set is linear, these conditions are both sufficient and necessary for global optimality. In the situations outlined by Johnson and Scheurman (*op. cit.*), solutions to the mathematical program representing the FMP can in some circumstances be obtained by direct application of the Kuhn Tucker conditions. In others, the conditions serve as the basis for algorithms to solve the FMP without having to perform the simplex procedure.

The model of Ware and Clutter (1971) is arguably the most famous model I formulation to be solved using a standard simplex procedure. The form of this model is indicated in Appendix 2.4.1.

2.4.2 Decomposition Methods Model I

All Forest Management Problems, when formulated as linear programs that have structural constraints that can be classified as model I or model II, have constraint matrices that are highly structured. These structural matrices arise because problem formulation requires the repeated specification of certain forms of constraints and variables. Consideration can be given to the structural properties of the constraint matrices when proposing solution techniques. The usual methods for doing this are termed *decomposition techniques* (Lasdon, 1970). Tchong (1966) (cited in Lasdon, 1970) proposed a Danzig-Wolfe decomposition method for the following harvest model.

$$\text{maximise } z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.35)$$

s.t.

$$u_j + v_j = b_j^{\max} - b_j^{\min} \quad \forall j \quad (2.36)$$

$$\sum_{i=1}^m r_i x_{ij} + u_j = b_j^{\max} \quad \forall j \quad (2.37)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \quad (2.38)$$

where

- r_i are units of area in management unit i
- b_j^{\min}, b_j^{\max} are minimum and maximum units of area that may be cut in year j
- c_{ij} in the yield from management unit i if cut in year j
- x_{ij} is the proportion of management unit i to be cut in year j , and
- u_j, v_j are slack and surplus variables for units of area to be cut in year j .

The decomposition procedure results in a subproblem that is easy to solve by inspection. This arises by placing all constraints of the form (2.38) in the subproblem. Tcheng's method has only one convexity row for the subproblem, it solves linear programs at the level of the subproblem by inspection and performs pivot operations at the level of the master to introduce the new column to the basis. These pivot operations ensure primal feasibility. Thus the slack variable constraints are satisfied (Lasdon, 1970).

Note that if Tcheng's methods were adopted to solve linear programs at the level of the subproblem and the master, by decomposing with m convexity constraints in the subproblem (the number of management units) and using the candidate solutions from each subproblem at the master level, then the master program resulting would have the same number of constraints as does the original problem, and so no reduction in computational effort would be achieved.

2.4.3 Decomposition Methods Model II

Recently, decomposition methods have been proposed for FMPs that possess model II structural constraints (Berck and Bible, 1984; Garcia, 1984). Of these proposed methods, that of Garcia's is more flexible because the model allows area transfers and multiple crop-types to be specified. Garcia's decomposition technique, therefore, is outlined in this section.

As with a model I decomposition, it is the structural constraints that form the constraint set for a model II subproblem. These are equations (2.12) through (2.15). These constraints were illustrated in Figure 2.3 for a single crop-type i and single age class j scheduled over six planning periods. It is very easy to envisage Figure 2.3 when more than one crop-type i is present. Furthermore, each crop-type may have a number of initial age classes. The resulting digraph may be converted into an acyclic network by appending a dummy source and dummy sink and breaking any cycles that may occur at points labelled (3) in Figure 2.3. Flows in this network will be area flows; moreover flows along each arc are uncapacitated, which means that the optimal flow in such a network will be the minimal cost flow.

Thus, solution of the subproblem in Garcia's formulation requires the identification of minimal cost paths in an acyclic network. This constitutes a well known network problem. A solution to such a problem can be obtained by constructing a solution to the dual of the network problem, and then identifying the solution to the primal problem from the optimal dual solution; the method is outlined in Wagner (1975). This solution will identify minimal cost paths for each of the initial areas in the initial crop-type/age

classes. The activity levels of arcs along these optimal paths can then be computed and the vector to enter the basis of the master program can be constructed. At the master level, a single pivot operation is performed at each iteration, there being only a single convexity row in the subproblem (Lasdon, 1970). The regulatory constraints are incorporated at the master level.

2.4.4 Dual Variable Estimation

Hoganson and Rose (1984) have proposed, in their paper entitled "A Simulation Approach for Optimal Timber Management Scheduling", a method that considers both the structural aspects of an FMP and the errors that may arise in specifying data input to FMPs. Although the title of the paper suggests a simulation approach, the problem is cast as a linear program, and a solution method is proposed that requires knowledge of certain dual variables. The simulation, if it may be termed that, involves the re-estimation of these dual variables at each iteration of an algorithm that endeavours to produce a possible primal solution.

The problem is specified by Hoganson and Rose (*op.cit.*) as

Problem P

$$\begin{array}{ll} \text{minimise} & \sum_{i=1}^I \sum_{j=1}^{J_i} c_{ij} x_{ij} \\ \text{over } & x \end{array} \quad (2.39)$$

$$\begin{array}{ll} \text{s.t} & \sum_{i=1}^I \sum_{j=1}^{J_i} v_{ijpt} x_{ij} = M_{pt} \quad \forall p, t \end{array} \quad (2.40)$$

$$\sum_{j=1}^{J_i} x_{ij} = A_i \quad \forall i \quad (2.41)$$

$$x_{ij} \geq 0 \quad \forall i, j \quad (2.42)$$

where

- A_i is the number of land units of stand type i present in the initial period;
- c_{ij} is the discounted cost of assigning a land unit of stand type i to management alternative j . This cost includes the value of an ending inventory;
- M_{pt} is the desired output level for product type p in time t ;
- I is the number of stand types;
- J_i is the number of management alternatives for stand type i ;
- v_{ijpt} is the per land unit yield of product type p in period t for stand type i , if management alternative j is followed; and
- x_{ij} is the number of land units of stand type i that are assigned to management alternative j .

The dual of the primal problem follows.

Problem D1

$$\begin{aligned} &\text{maximise} && \sum_{t=1}^T \sum_{p=1}^P M_{pt} m_{pt} && \sum_{i=1}^I A_i a_i && (2.43) \\ & && m, n \end{aligned}$$

$$\begin{aligned} &\text{s.t} && \sum_{t=1}^T \sum_{p=1}^P v_{ijpt} m_{pt} - a_i \leq c_{ij} && \forall i, j && (2.44) \end{aligned}$$

$$a_i \text{ unsigned} \quad \forall i \quad (2.45)$$

$$m_{pt} \text{ unsigned} \quad \forall p, t \quad (2.46)$$

where

m_{pt} is the dual variable associated with the output level constraint for product p in time t ;

a_i is the negative of the dual variable associated with stand type i ;

P is the number of product types; and

T is the number of time periods.

Hoganson and Rose (*op. cit.*) provide the following interpretation of the dual.

"An outside party wishes to purchase all the land from the landowner and in return, sell to the landowner the landowner's desired outputs. The problem of the outside party is to determine the price to charge for outputs in each period (m_{pt} 's),

and the price to offer for each stand type (a_i 's) so that the outside party's returns are maximised. The outside party is constrained in that the price offered for each stand type must make it profitable for the landowner to sell that stand type rather than manage it."

A facile solution technique would result if the dual variables, the m_{pt} 's in problem D1, were known, the dual problem would then become the following problem.

Problem D2

$$\begin{array}{ll} \text{minimise} & \sum_{i=1}^I A_i a_i \\ a & \end{array} \quad (2.47)$$

s.t.

$$a_i \geq \sum_{t=1}^T \sum_{p=1}^P v_{ijpt} m_{pt} - c_{ij} \quad \forall i, j \quad (2.48)$$

$$a_i \text{ unsigned} \quad \forall i \quad (2.49)$$

Thus is just the problem of minimising the land purchase cost for the outside party. This problem is very easy to solve because all constraints but one must be redundant for each a_i . This will be the maximum of the right-hand side of (2.48) for each a_i . Such a problem can be solved by inspection.

The proposed algorithm re-estimates the dual variables (m_{pt} 's) at each iteration by taking into account relationships that exist between the output levels and the marginal costs of production (prices to charge for output for the outside party). The algorithm terminates whenever a set of multipliers (m_{pt} 's) is found that is dual feasible and close to dual optimal, that is, primal optimal and close to primal feasibility.

The algorithm is as follows:

- (1) Use prior information about the problem to estimate the marginal costs of production for each output and period (m_{pt} 's).
- (2) Assume the m_{pt} estimates are correct and solve problem D2 for the remaining dual variables (a_i 's).
- (3) Determine a primal solution (the x_{ij} 's in Problem P) that correspond to the optimal dual solution. This primal solution is not necessarily a feasible solution.
- (4) Determine the output levels for the primal solution found in Step (3) so that the primal solution can be tested for feasibility.
- (5) Test for primal feasibility. If the output levels determined in Step 4 are close to their desired levels (m_{pt} 's), stop, the primal solution is both an optimal and near-feasible solution.

Otherwise perform Step (6).

- (6) Use the output levels determined in Step 4 and a basic understanding of the relationship between output levels and marginal costs of production to re-estimate the m_{pt} values.
- (7) Return to Step (2).

This procedure is possibly of importance as a solution technique for FMPs in that it considers the wide range of near feasible solutions. Computational experience indicates a large number of basic solutions in the neighbourhood of an optimal solution for an FMP (Whyte pers. comm.). This technique identifies those basic solutions that are non-feasible to the primal problem P. Should primal feasibility be reached, the algorithm will terminate by Step (3) above. Furthermore, this procedure acknowledges from the outset the possible imprecision and inaccuracy of data items that are used within the model structure, since the computational work required to identify solutions that are close to primal feasibility is greatly reduced from that required to identify the optimum to a problem, possibly inaccurately specified, by deterministic simplex methods.

The future success of this technique will rest on the ability to accurately estimate the dual variables (m_{pt} 's) at each iteration of the algorithm above. Hoganson and Rose (*op. cit.*) describe methods by which this can be done. However, they identify the procedure currently being used as being "crude".

2.5 REGULATORY CONSTRAINTS IN FOREST UTILISATION MANAGEMENT PROBLEMS

Regulatory constraints, as indicated in section 2.3, can be specified for a wide variety of resource sets. However, an important consideration in their imposition is whether the problem being solved is an FMP or a FUMP. In an FMP, one may need to impose regulatory constraints to smooth the flow of forest resources at the level of the processes that produce those resources. In a FUMP, part of this smoothing may be undertaken at the level of processing centres that utilise forest resources. The regulatory constraints imposed at forest level in a FUMP may possibly be highly aggregated, for example, smoothing and bounding constraints for total harvest volume. Thus, the production and regulation of forest resources may be determined by the integrated activities of processing centres representing forest growth and centres representing the utilisation of forest resources. Contrast this with the situation where the flow of forest resources in a FUMP is achieved by the imposition of regulatory constraints at the forest level only. Here, the pattern of resource utilisation may be unduly constrained by supply, and the economic efficiency of the system governed by this supply.

This notion of regulating processing centres through integration and allowing them to determine their operational status jointly is discussed further in chapter 3, where the transportation, utilisation and marketing of forest based resources are considered.

CHAPTER 3

TRANSPORTATION, UTILISATION AND MARKETING

This chapter is concerned with the control of resource flows within a modelled system representing forests and utilisation management alternatives. The flows span transport from forests to markets for produce both with and without further processing. These flows are initiated by the removal of roundwood from management units within forests. The mechanisms to govern these removals were discussed in Chapter 2. Thus, the regulatory devices in a FMP constitute a means to govern roundwood supplies over time. Generally, subsequent processing alternatives are dependent on well modulated roundwood supplies.

Alternative mechanisms to control the resource flows occurring within a FUMP are developed and explained in this chapter. Central to this development is the formulation of mechanisms to represent some standard Operational Research problems that occur jointly within a FUMP. These problems arise in the control of resources flowing to and from *processing facilities*,¹ although each problem may appear in a slightly modified form from the generally recognised pure Operations Research forms. Resource flow control problems include the problems of: location; resource allocation; capital budgeting; and fixed charge specification.

A modelling device enabling the control of resources for a processing facility shall, for the purposes of this study be termed a *regulatory mechanism*. When extra capacity has been made

¹ The term processing facility is taken to be a specialised form of processing centre where resources produced in forests are utilised. Amongst its inputs are roundwood (logs) and output contains roundwood conversion products. Examples would include sawmills, pulpmills, and integrated wood processing plants.

available at a processing facility an *introduction* is considered to have occurred. Thus "introduction regulation mechanisms" are devices that regulate resource flows and also aspects associated with introduction of capacity. It is the development and discussion of introduction regulation mechanisms that constitutes the largest part of this chapter. The need to formulate and examine mechanisms where introductions occur arises quite naturally in Forest Utilisation Management Problems that consider possible forms of utilisation of roundwood having either non-existent or insufficient processing capacity. For example, consider the feasibility of expanding sawn timber production during the model time frame by adding extra capacity at a sawmill. This requires that consideration be given to regulation of resources to and from the sawmill, along with the possibility of utilisation of the extra capacity.

The introduction regulation mechanisms developed are quite generalised and can be used to control processing activities of any facility where the feasible production alternatives are specified through a set of linear variables and linear constraints. In Chapter 5, an application of introductory regulatory mechanisms involving binary variables is given, in which the processing facilities considered are sawmills of differing capacities.

This chapter also identifies the importance of being able to define and adequately measure the flows that occur. This is borne out in a discussion centred almost entirely on the need for adequate log grading systems. Discussion on this is given because roundwood flows are possibly the most ill-defined flows within a modelled system representing forests and associated utilisation alternatives. Generally,

processed roundwood is more amenable to description because processing induces homogeneity to various degrees, which can be distinguished by application of a resource grading system.

The resource flows within a modelled system representing a FUMP are best envisaged by construction of a *general flow digraph*, an example of which is shown in Figure 3.1. This indicates any possible flows or transfer of resources between processing centres during the model time frame.

Steps 1 through 7 below constitute a generating procedure for digraphs structured as that in Figure 3.1. Note that nothing is said about what is produced or consumed at processing centres: the digraph merely indicates possible transfer of resources and supplies limited information about the spatial distribution of processing centres.

1. Identify the set of forests that may exist during the model time frame (planning horizon); denote this by the set of vertices $\{v_p : p=1, \dots, n_p\}$.
2. Identify the set of processing facilities that may exist during the model time frame; denote this by the set of vertices $\{v_q : q=1, \dots, n_q\}$.
3. Identify the set of markets that may exist during the model time frame; denote this by the set of vertices $\{v_r : r=1, \dots, n_r\}$.
4. Construct arcs between forests and markets by the following rule (these arcs represent transfer of roundwood to log sale points); for each forest $v_i \in \{v_p : p=1, \dots, n_p\}$, and for each market $v_\ell \in \{v_r : r=1, \dots, n_r\}$, construct the arc (v_i, v_ℓ) if forest v_i may ship resources to log sale market v_ℓ .

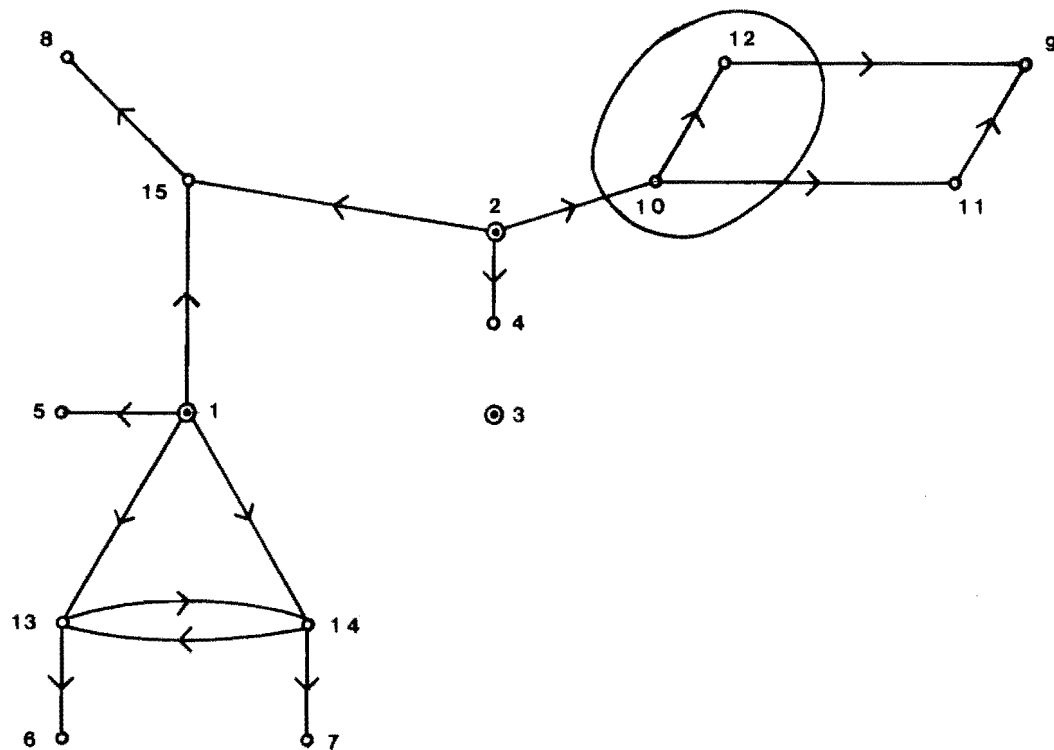


Figure 3.1 : General Flow Digraph

Arcs indicate possible transfer of resources between processing facilities, represented as vertices. Vertices labelled 1, 2, and 3 having no arcs entering, are forests; while those labelled 4, 5, 6, 7, 8, and 9, are markets (no arcs leaving). Vertices with arcs both entering and leaving, labelled 10, 11, 12, 13, 14, and 15 represent processing facilities. The presence of the isolated vertex 3 corresponds to a young forest, which does not give rise to recoverable volume in the model time frame. Processing facilities at the same location may be indicated by enclosing the vertices concerned with a simple closed curve as in vertices labelled 10 and 12 above.

5. Construct arcs between forests and processing facilities by the following rule: for each forest, $v_i \in \{v_p : p=1, \dots, n_p\}$ and for each processing facility $v_j \in \{v_q : q=1, \dots, n_p\}$. Construct the arc (v_i, v_j) if forest v_i may ship resources to facility v_j during the model time frame.
6. Construct arcs between processing facilities by the following rule: for each pair of processing facilities, $v_j, v_k \in \{v_q : q=1, \dots, n_p\} j \neq k$. Construct the arc (v_j, v_k) if facility v_j may ship resources to facility v_k during the model time frame. Enclose facilities at common locations by a simple closed curve.
7. Construct arcs between processing facilities and markets by the following rule: for each processing facility $v_j \in \{v_q : q=1, \dots, n_q\}$ and for each market, $v_\ell \in \{v_r : r=1, \dots, n_r\}$. Construct the arc (v_j, v_ℓ) if facility v_j may ship resources to market v_ℓ during the model time frame.

From the viewpoint of the general flow digraph the problem of a FUMP is to provide detail on the processing alternatives at each processing centre in terms of resources consumed and produced and to integrate flows between centres in such a way that the overall system is efficient according to some economic criterion. Flows between centres possess both temporal and spatial attributes: temporal because production of roundwood by management units can occur at different times; and spatial because of possibly different geographic locations of processing centres within an FUMP system.

Actual flows in the real system depend on such factors as forests being old enough to produce roundwood resources,

processing facilities being capable of consuming and producing resources, the resource sets being consumed and produced at processing centres, and constraints on production at processing centres.

This chapter deals with methods that may be used to regulate flows within a modelled system representing a FUMP. Generally, these flows must be adequately defined, in terms of both composition and measurement, which topic is the basis of the discussion in section 3.1. Flows of resources between centres are required to ensure their integration; transportation aspects are described in section 3.2. Developed and discussed in section 3.3 are introductory regulatory mechanisms that are required for the introduction of capacity at processing facilities. Special attention is paid to non-convex mechanisms that require binary variables. Marketing mechanisms are introduced and discussed in section 3.4, which includes examination of the possibility of using stochastic programming techniques to take account of stochastic elements that may arise in markets. Finally, section 3.5 proposes solution techniques that account for structural aspects within FUMPs.

3.1 DESCRIPTION AND MEASUREMENT OF RESOURCE FLOWS

Efficiency in integration of processing centres is coordinated through use of an economic efficiency measure, which in turn must incorporate measures of flow where either revenue is generated or costs are incurred. All flows within a modelled system representing the extraction of wood from forests and transfer to markets with the possibility of further processing must be capable of definition and measurement. Well defined grading systems exist for processed

roundwood, although, in some areas, there is some duplication in the form of alternate systems. For example, both machine and visual grading rules for sawn lumber currently operate in New Zealand (Vaney, J.C., 1981). Processed roundwood is generally easier to grade than roundwood in that all currently known forms of processing involve the initial breakdown of roundwood, and the possible subsequent reassembly into resources that have a higher degree of homogeneity associated with them. These processes of breakdown and subsequent reassembly are undertaken by biological, chemical or mechanical means. The description and measurement of resource flows other than roundwood flows are not considered further in this study.

One procedure used to describe roundwood (logs) is known as a log grading system. Its essential feature is that it constitutes a procedure to sort logs into various classes such that each class is relatively homogeneous according to some defined criteria (namely, log grading rules), yet with recognisable differences between classes. Class boundaries are chosen in such a way as to identify commercially important resources. When classes do not overlap any log can be allocated to only one log class (mathematically a log grading system identifies a partition), but can have a hierarchy of possible uses.

To be of benefit a log grading system must be able to classify roundwood for all intended forms of utilisation. Different forms of utilisation require descriptions using possibly different sets of attributes. A sufficient number of attributes must be employed so as to be able to differentiate between the various input requirements of the

different utilisation modes.

Typically, the log is considered as a basic unit that is further described by length, diameter (usually small end diameter, sed); taper (meaning diameter change per unit length), internodal length or equivalently the maximum number of whorls per unit length, straightness or measures or sweep and crook, and other quality characteristics such as maximum knot size.

Besides the basic log, collections of logs can also be classified. These define a log set by specification of *mix limits*. Typically, mix limits are defined in terms of some or all of the attributes that are used to classify individual logs. These attributes must be readily quantifiable because implementation of log grading rules for large numbers of logs requires that measurements be made quickly. As an example of mix limits used to define input requirements for a processing facility, consider a modern sawmill designed to process logs of specified dimensions. Mix limits for the facility could be stated as proportions of total log volume input that may occupy various diameter and length classes.

The attributes used to classify an individual log or log set may be regarded as the components of a random vector, each component being either discrete or continuous. The determination of classes into which logs are sorted corresponds to the identification of sets in the domain of the random vector (Mood *et al.*, 1974; Zehna, 1970).

Log grading systems play an important role in modelled systems representing FUMPs in that the description of roundwood removed from forests is made according to grading rules. The number of classes into which logs are sorted affects both the level of aggregation and the accuracy of representation within the modelled system. If more classes are used,

then lower levels of aggregation are obviously achieved. This is accompanied by a higher accuracy of representation and a reduction of the stochastic effects introduced into the model structure. The reduction occurs because the domain of the joint random vector used to classify the roundwood resource is partitioned into a greater number of sets. Each of these sets is identified by a finite number of parameters such that further processing of a set is described in terms of its identifying parameters. The process of increased accuracy in representation is entirely analogous to a discrete approximation to a continuous univariate distribution using partitions of decreasing mesh size; each partition induces sets in the domain of the random variable, and as the mesh size is decreased, the number of sets induced increases.

Achieving reductions in aggregation and stochastic effects in a modelled system representing a FUMP must be balanced against the requirements of extra variables and constraints required in a model structure, and also, against the required extensions to the model data base.

It would seem sensible, then, that a log grading system be determined that could apply to all log sales. This would then serve to define the material actually exchanged at a point of sale. This in fact, was one of the objectives of a joint industry/Forest Research Institute (FRI) study, the Conversion Planning Project, set up in New Zealand in 1982 (Doyle pers. comm., 1984). However, the adoption of log grading rules has met with considerable opposition from some concerned parties, so much so, that log sales are conducted largely on an *ad hoc* basis, with different grading rules being applied for different log sales.

Perhaps some understanding as to why this situation currently prevails can be gained by considering the domestic and export prices of sawlogs of the same grade. Richardson (1985), formerly a director of the New Zealand Forestry Council, stated

"Since 1977, export sawlog prices to the Forest Service have only once averaged less than twice the domestic price - in real dollars, that difference is highly significant".

This price differential is indicative of a general divergence of interest between forest growers and utilisers of roundwood in New Zealand. Whyte (pers. comm.) identifies the presence of commercial interests as a contributory factor in failure to adopt standardised log grading rules. This failure may, in the short term, convey protection to domestic processors by way of reduced prices for log inputs, while in the longer term it could jeopardise the ability of the domestic forest-based industry to face external competition thereby lowering the ability of that industry to attract investment capital and raise doubts as to its overall commercial viability (Richardson, 1985; Smyth, 1985).

A log grading system may be incorporated with a bucking mechanism, this approach may be taken when classifying recoverable volume from forests within a representation of a FUMP. Resulting from this, is the distribution of recoverable volume amongst roundwood classes of differing qualities. The test problem discussed in Chapter 5 involved a classification of recoverable volume in this way, a description of the procedure is given in the paragraph entitled Roundwood Bucking Mechanism in subsection 5.1.1.

3.2 THE TRANSPORTATION MECHANISM

Unlike the widely recognised transportation problem introduced by Hitchcock (1941) (cited in Gass (1975)), expanded by Koopmans (1949) (cited in Gass (1975)) which may be solved using either the simplex method or related algorithms that account for the structure of the problem, notably the duality relations (see Gass, 1975), the transportation mechanism within a forest utilisation management problem is simply a linkage mechanism between sets of joint productive processes (that is, processes where product formation requires more than one activity). The links are provided in such a way that the identity of the commodity is clearly defined, as are the locations between which the commodity is shipped and the time of shipping. Further, conservation of flow must be ensured in these links. The transportation mechanism simply represents the transfer of resource sets between processing centres.

A complication arises in multi-time period models when introductions of processing facilities that do not exist prior to the planning horizon are considered. In this case, a facility is not capable of processing resources received via transportation links until such time as capacity is allocated and production at the facility commences (that is, until an introduction has occurred and a positive capacity is made available to the facility), nor is the facility capable of shipping resources produced to sites that may receive them. Thus, shipment of resources to and from facilities can be undertaken only when those facilities are operational. At that stage, resources are being both consumed and produced.

This complication is dealt with by making the transportation mechanism operate in conjunction with the mechanism that makes capacity available at a facility, namely an introduction mechanism which is the subject of section 3.3. The availability of capacity at a facility, then, means that arcs to and from the facility may be used to transport resource inputs and outputs respectively. The converse, namely non-availability of capacity, is associated with arcs not being used. The constraints that may be used for the transportation mechanism are indicated in section 3.3.

3.3 INTRODUCTORY REGULATORY MECHANISMS

The regulation of resource flows to and from processing facilities is the subject of this section. Specifically, mechanisms to regulate these flows for facilities that may have capacity introduced at any time during a multi-period planning horizon are developed and examined. These mechanisms are either non-convex, that is, they do not give rise to a convex programming problem (Gass, 1975), and require the use of binary variables to resolve the non-convexity, or else they are convex, not requiring the use of binary variables. However, as will subsequently be shown in this section, convex mechanisms impose certain limitations as to how capacity increments may be modelled.

Note that several physical situations could correspond to an introduction. From a modelling viewpoint, these situations are largely of interest only in terms of data collection in that they can all be modelled using the same model formulation.

Typically, an introduction is consistent with any of the following:

1. Physical capital formation, to allow either construction of the facility or additions to an existing facility.
2. The direct purchase of existing physical capital, thereby making capacity available.
3. A leasing arrangement to make capacity available to a lessee.

In each case, from a modelling viewpoint, the problems faced are largely the same, but from an accounting viewpoint, this is not necessarily so. For example, 1 above may involve loan repayments, whereas 3 involves payment on a lease, in either case a form of payment is made for the utilisation of processing capacity.

In introducing the mechanisms to regulate flow, particular attention is given to the non-convex techniques, which do not require limiting consumptions to be made about capacity additions. Non-convex mechanisms do, however, require integer programming techniques.

Integer programs have assumed an increasing importance since the introduction of the simplex procedure in 1947². Important and wide-ranging classes of problem can be formulated using integer programming techniques. A historical summary of well known integer programming problems is provided by Gass (1975). Williams (1978) identifies special classes of integer programs as being set covering, set packing, or set partitioning problems. The use of integer variables proposed in this study could be termed a set selection problem, in that, from a set of possible capacity introductions, part

² The simplex procedure was first developed by Danzig in 1947, although Kantorovich (cited in Gass, 1975), a Soviet mathematician and economist, formulated and solved a linear program in 1939.

of a FUMP problem is to determine which, if any, introductions are to be made.

To date, solution mechanisms can be classified as either cutting plane or enumerative techniques. The former, initially proposed by Gomory (1958) to solve pure integer problems, and subsequently extended by Benders (1962) to solve mixed integer problems, fathoms the problem using a relaxation technique that progressively defines the convex hull of feasible integer solutions by addition of further constraints (Dallenbach *et al.*, 1983). Enumerative techniques, on the other hand, are implicit in that they investigate only a restricted subset of a possibly large set of integer solutions, requiring any optimal solution to be contained in this subset. The means of performing the systematic examination distinguishes the techniques. The most widely used enumerative integer programming techniques are the Branch and Bound methods, originally proposed by Land and Doig (1960). These can be applied to either pure-integer or mixed-integer situations (inclusive of mixed integer linear programs). Branch and Bound techniques are best thought of as not isolated methods for separate problems but as a family of related techniques united by a common approach. For mixed integer linear programs branch and bound techniques solve the problem by a combination of "relaxation" and "separation" with "fathoming" of the problem being achieved through the subproblems (that is, the descendants) that arise through "separation" (Dallenbach *et al.*, 1983; Murty, 1976). The additive algorithm proposed by Balas (1967) is also an enumerative technique. Subsection 3.5.1 of this chapter discusses application of the cutting plane technique proposed by Benders (1962) to FUMPs with non-convex introduction mechanisms, while the test problem

of Chapter 5 involved an application of a branch and bound procedure.

3.3.1 Non-Convex Interpolating Binary Mechanism

The introduction and regulation mechanisms developed in this subsection are non-convex and they require the use of integer programming techniques to resolve the non-convex programming problem that arises (Gass, 1975; Taha, 1976). In addition these mechanisms are interpolating in the sense that they allow the modelling of introductions at all times during a model period, and may be used to model introductions at all times during a planning horizon. Contrast this with the situation where introductions are permitted only at a specified point in a model period. Such methods can be used only to model introductions at selected times during a planning horizon (namely, at one time point in each model period). Introduction and regulation mechanisms that operate in this way are developed in subsections 3.3.2 and 3.3.3.

The development of the non-convex interpolating binary mechanism is initially done with respect to a single arbitrary facility at an arbitrary location. This requires the definition of the set S_1 , which is a set of continuous weights $(\omega_{2t-1}, \omega_{2t})$, the values for which are governed by a controlling set of binary variables (y_t) . The constraint system (3.1) specifies a relationship amongst the elements of S_1 . This system, and its analogues developed in subsections 3.3.2 and 3.3.3, constitute the most important systems in developing introduction regulation mechanisms.

$$S_1 = \{y_t, \omega_{2t-1}, \omega_{2t} : y_t = 0 \text{ or } 1; \omega_{2t-1}, \omega_{2t} \geq 0; t=1, \dots, n\}$$

$$\omega_{2t-1} + \omega_{2t} = y_t \quad \forall t \quad (3.1)$$

where

t is the subscript denoting model time period.

Each binary variable y_t , has associated with it two continuous positive weights, ω_{2t-1} and ω_{2t} . Figure 3.2 illustrates the temporal relationship between the weights and the binary variables. The weights ω_{2t-1} and ω_{2t} are associated with the start and finish of model period t respectively. Moreover, the weights ω_{2t} and $\omega_{2(t+1)-1}$ are considered to represent a coincident point on a time scale, namely the end of period t and the start of period $t+1$.

planning period	binary variable	start	weight	finish
1	Y_1	ω_1		ω_2
2	Y_2	ω_3		ω_4
3	Y_3	ω_5		ω_6
.	.	.		.
.	.	.		.
.	.	.		.
n	Y_n	ω_{2n-1}		ω_{2n}

Figure 3.2 : Control of continuous weights

Each planning period involves two continuous weights and a controlling binary variable. The oblique lines link weights associated with the same point on a time scale.

For an introduction in period t , then y_t has the value 1, the system 3.1 ensures one of the following cases will arise.

Introduction Cases

- Case I1 The introduction occurs at the beginning of planning period t , then $\omega_{2t-1} = 1$ and $\omega_{2t} = 0$.
- Case I2 The introduction occurs during planning period t , not at its end points, then $\omega_{2t-1} > 0$, $\omega_{2t} > 0$ and $\omega_{2t-1} + \omega_{2t} = 1$.
- Case I3 The introduction occurs at the end of planning period t , then $\omega_{2t-1} = 0$ and $\omega_{2t} = 1$.

These correspond respectively to an introduction at the beginning, during, and the end of a planning period.

Subsections 3.3.1.1; 3.3.1.2; 3.3.1.3; and 3.3.1.4 resolve the problems of location; resource allocation; capital budgeting; and fixed-charge specification respectively that arise with an introduction. Central to this resolution is the set S_1 and the constraint system (3.1).

3.3.1.1 The Location Problem

A location problem arises when a processing facility of a specified type (sawmill, pulpmill and so on) may be introduced at more than one location, and the "best" location(s) must be identified (Dallenbach *et al.*, 1983). The pure location problem equates directly to minimisation of transport costs on arcs to and from the location when facilities receive the same input sets, have the same production function, and produce the same output sets. When any of these assumptions are violated, as they may be within FUMPs, then location alone will not determine the optimal form of integration of processing centres.

Regardless of whether the location problem is in a pure form, if a facility is to be allowed the possibility of introduction at several locations, the full production model for the facility, along with accompanying introduction and regulation mechanisms can be introduced into the model for each location. The location problem will then be solved during model solution as a part of determining the most efficient way to integrate the activities of processing centres.

The above discussion suggests that, if the location problem is to be incorporated as part of the introductory regulatory mechanism, then the set S_1 must be modified as follows

$$S_{1p} = \{y_{pt}, \omega_{p2t-1}, \omega_{p2t}; y_{pt} = 0 \text{ or } 1; \omega_{p2t-1}, \omega_{p2t} \geq 0; t=1, \dots, n\}$$

$$\omega_{p2t-1} + \omega_{p2t} \leq y_{pt} \forall t$$

where

t is the subscript denoting model time period, and

p is the subscript denoting location and type of facility.

The modification merely indicates an introductory regulatory mechanism for each facility at each location where capacity introductions may occur. For ease of exposition, the additional subscript p will most often be dropped in the subsequent development. Only when summations over p occur will the subscript be explicitly included.

3.3.1.2 Resource Allocation Problem

The resource allocation problem arises when considering capacity introductions at a processing facility. Capacity made available allows the consumptions of inputs and the production of outputs. Conversely, no utilisation may take

place when capacity is not available. The constraint systems necessary to deal with the resource allocation problem are developed in this subsection.

The seemingly realistic approach adopted, is that, should an introduction occur at any point in a model period except as its finish point (as in cases I1 and I3 in subsection 3.1), then resources will be made available to the processing facility for that period in direct proportion to the length of time it has been operating in that period. Thus, should an introduction occur at the end of the model (case I3 in subsection 3.1), no resources will be made available during that period. Additionally, if an introduction has occurred, then at all subsequent model periods in the planning horizon, if they exist, the facility must be capable of receiving its full complement of resources.

$$\sum_{s=1}^t r_{is} \omega_{2s-1} + \sum_{s=1}^{t-1} r_{is} \omega_{2s} \quad \forall i, t \quad (3.2)$$

$$\sum_{s=1}^t r_{is}^* \omega_{2s-1} + \sum_{s=1}^{t-1} r_{is}^* \omega_{2s} \quad \forall i, t \quad (3.3)$$

$$i=1, \dots, I$$

$$t=1, \dots, n$$

where

r_{is} (r_{is}^*) are the lower (upper) limits for the consumption or production of resource i in period s by a specified processing facility.

Consider the expressions (3.2) and (3.3), should an introduction occur before period t , say t' , then (3.1) ensures that $\omega_{2t'-1} + \omega_{2t'} = 1$. If this is the only introduction, then no other weights from S_1 are positive. Expressions (3.2) and (3.3) will then respectively have values $r_{it'}$ and $r_{it'}^*$, so

that (3.2) and (3.3) correctly sum to the lower and upper bounds of resource i in period t' . However should an introduction occur during period t , then by the previous argument, it should receive resources in period t only for the length of time it is in operation. This corresponds to using only the initial weight of the pair $(\omega_{2t-1}, \omega_{2t})$ for resource allocation during period t . Thus for an introduction in period t , with no previous introductions, (3.2) and (3.3) consist of the non zero terms $r_{it}\omega_{2t-1}$ and $r_{it}^*\omega_{2t-1}$ respectively.

The system (3.1) and expressions (3.2) and (3.3) are also consistent when more than one introduction of specified facility has occurred prior to period t . Consider multiple introductions prior to period t (that is, more than one $y_{t'} = 1$ for $t' < t$, $y_{t'} \in S_1$). Table 3.1 shows values generated by expressions (3.2) and (3.3), assuming $r_{it}(r_{it}^*)$ are constant for all t .

When considering multiple introductions of the same facility at the same location, it is sufficient to consider the production model for a single facility and to modify it in the following way. Each time an introduction occurs, ensure that the resource bounds for all resources either produced or consumed by the facility are adjusted to the levels required by the number of facilities operating (as indicated in Table 3.1). The processes (activities) within the production model can then consume or produce resources within the newly specified limits. This approach is only possible because the production models are represented in a linear fashion. It is an application of the principle of linear superposition, a principle that is more usually associated with solutions to linear differential equations, but which is

Table 3.1 : Tabulation of Resource Bounds

Shown are the lower and upper bounds for resource i , as a function of the number of introductions prior to period t . The facility is assumed to produce or consume resource i between the bounds r_i and r_i^* during any period it is in operation

Introductions prior to period t	Resource i availability in period t	
	Lower bound	Upper bound
1	r_i	r_i^*
2	$2r_i$	$2r_i^*$
3	$3r_i$	$3r_i^*$
.	.	.
$t-1$	$(t-1)r_i$	$(t-1)r_i^*$

applicable to any linear system (Rainville and Bedient, 1969).

This means that the activities associated with introductions of a processing facility of specified type at a specified location can be represented by including the production model once in each period and using the system (3.1) through (3.3) to perform the resource allocation by adjusting bounds in any period according to the number of facilities operating.

Besides bounding resources between the correct levels, resources that are being consumed must be regulated in processes that utilise them. Similarly, resources must be regulated in processes where they are produced. When a resource is produced, regulation may be achieved by forming the inequalities (3.4) and (3.5) from expressions (3.2) and (3.3) respectively. Formulation of a constraint set

for a resource that is consumed and for a resource that is both produced and consumed is not difficult. However, in the latter case, the relationship between net production and net consumption of the resource must be known.

$$\sum_{s=1}^t r_{is} \omega_{2s-1} + \sum_{s=1}^{t-1} r_{is} \omega_{2s} - \sum_{k \in K_i} a_{ki} x_{tk} \leq 0 \quad \forall i, t \quad (3.4)$$

$$\sum_{s=1}^t r_{is}^* \omega_{2s-1} + \sum_{s=1}^{t-1} r_{is}^* \omega_{2s} - \sum_{k \in K_i} a_{ki} x_{tk} \geq 0 \quad \forall i, t \quad (3.5)$$

$$i = 1, \dots, I$$

$$t = 1, \dots, n$$

where

x_{tk} is the activity level of process k in period t , $x_{tk} \geq 0$;

a_{ki} is a technological coefficient indicating production of resource i per unit activity of process k ; and

K_i is the set of processes for the processing facility that produce resource i .

The system comprising (3.1), (3.4) and (3.5) constitute the resource allocation mechanism and may be used to regulate the flow of resources from a processing facility between any desired bounds (note the analogous situation for resources consumed). Systems similar to (3.4) and (3.5) can be developed to distribute resources to and from transportation arcs which link a facility with other processing centres

To consider this, assume that introductions of facilities (one or more) at location ℓ that produce resource i have been made prior or during period t . Then during period t , resource i must be transported from the facilities at location ℓ to other locations that have processing centres that can utilise resource i as part of their input. The constraint system (3.6) is then required.

$$\sum_{p \in P_\ell} \sum_{k \in K_{ip}} a_{pki} x_{ptk} - \sum_{q \in L_{i\ell t}} u_{\ell qti} = 0 \quad \forall i, t \quad (3.6)$$

where

- P_ℓ is the set of processing facilities that may be introduced at location ℓ ,
- K_{ip} is the set of activities for processing facility p that produce resource i ,
- $L_{i\ell t}$ is the set of destinations to which resource i may be shipped in period t when the originating location is ℓ ,
- a_{pki} is the technological coefficient indicating production resource i , per unit activity of process k for facility p ,
- x_{ptk} is the activity level of process k in period t for facility p using introduced capacity, and
- $u_{\ell qti}$ are the units of resource i shipped from location ℓ to location q in period t .

The left-hand summation in (3.6) merely collect resource i from the activities at facilities at location ℓ that provide it, while the right hand summation distributes this cumulative resource into arcs that terminate at destinations where the resource may be used in subsequent processing.

Similarly, when resource j may possibly be utilised at location ℓ , then at any period t , all arcs carrying resource j and ending at ℓ must be summed over, the system (3.7) results.

$$\sum_{q \in M_{j\ell t}} u_{q\ell tj} - \sum_{p \in P_\ell} \sum_{k \in K_{jp}} a_{pkj} x_{ptk} = 0 \quad \forall j, t \quad (3.7)$$

$$j = 1, \dots, J$$

$$t = 1, \dots, n$$

where

- $M_{j\ell t}$ are the locations from which resource j may be shipped to location ℓ in period t , and
- K_{jp} is the set of processes for facility p that consume resource j .

Thus, the systems (3.6) and (3.7) in conjunction with (3.1), (3.4) and (3.5) represent transportation and resource allocation mechanisms. It is important to note (3.6) and (3.7) will function correctly provided (3.1), (3.4) and (3.5) function correctly, i.e., capacity introduced at facilities can neither receive nor ship resources until such time as processes utilising that capacity become operational.

3.3.1.3 The Budgeting Problem

The introduction of capacity at a processing facility is associated with costs to the modelled system. These represent payments that must be made in order to acquire the capacity. Limits may exist on payments that can be made in any or all model periods. Similarly, limits may exist on total expenditure by way of payments. The imposed limits depend on such factors as the availability of finance; interest rates on loans or mortgages secured; the willingness of the forest-based company to face these payments. The budgeting problem is to ensure that introductions that occur are financially feasible. As will be seen in this subsection, budgeting results in a series of constraints involving weights from the set S_1 . The inclusion of cost measures associated with introduction into the objective function is the subject of the next subsection. When introductions are non-convex, this becomes the familiar fixed-charge problem.

Payments associated with an introduction may be made over one or more model periods, which suggests that, to introduce facility p at period t , there will be a sequence of payments made to meet the cost incurred with introduction. Not all these payments need fall in the model time frame.

Generally, they will start with the initial period of introduction and continue until the facility is paid-off or until the final model period, whichever comes sooner. The sequences (3.8), one for each possible period of introduction, result.

$$\left\{ C_{ptq} \right\}^{\min[n, t+m_p-1]} \quad \forall t \quad (3.8)$$

$$t = 1, \dots, n$$

where

C_{ptq} is the payment incurred in period q in order to introduce facility p at the start of period t ,
 $C_{ptq} > 0$.

m_p is the number of model periods over which payments are made for facility p , assuming the first payment is coincident with introduction.

The definition of payments in this way means that only payments occurring during the model time frame are considered within the budgeting mechanism. Payments incurred beyond the planning horizon are purposefully omitted. This is why the minimum is specified in (3.8). The sequences defined by (3.8) are of sufficient interest to tabulate an example, which is given in Table 3.2. The sequences show a temporal pattern determined by the introduction period, the repayment period, and the number of model periods. For simplicity, the indice p is dropped in the tabulation.

Expressions for the total payments that may be incurred during period t can be constructed from the set of sequences (3.8), together with the continuous weights from the set S_1 , as in (3.9). The terms involving $\omega_{p,2(s-1)}$ correspond to introductions that occur at the end of period $s-1$, while terms involving $\omega_{p,2s-1}$ arise from introductions that occur at the start of period s . These two points are co-incident in time. The introduction times considered

Table 3.2 : Tabulation of Repayment Sequences

	$\begin{array}{cccccccccccc} & & & & & & & & & & \text{---}Y_6\text{---} \\ & & & & & & & & & \text{---}Y_5\text{---} \\ & & & & & & \text{---}Y_4\text{---} \\ & & & & \text{---}Y_3\text{---} \\ & & \text{---}Y_2\text{---} \\ & \text{---}Y_1\text{---} \end{array}$											
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}	ω_{12}
1	C_{11}											
2	C_{12}	C_{22}	C_{22}									
3	C_{13}	C_{23}	C_{23}	C_{33}	C_{33}							
4	C_{14}	C_{24}	C_{24}	C_{34}	C_{34}	C_{44}	C_{44}					
5		C_{25}	C_{25}	C_{35}	C_{35}	C_{45}	C_{45}	C_{55}	C_{55}			
6				C_{36}	C_{36}	C_{46}	C_{46}	C_{56}	C_{56}	C_{66}	C_{66}	

Shown are repayments sequences (columns) possibly incurred during a six-period planning horizon, assuming that repayments span four periods. At the left-hand side of each row is the model period index. At the head of each column is the continuous weight associated with the introduction mechanism. Each Table element is of the form C_{tq} = payment incurred at the start of period q in order to introduce a facility at the start of period t . Above the Table, each horizontal line, enclosing a y_i indicates the binary variable associated with the weights below the line

may not include all planning periods, in which case, not all terms in (3.9) will be defined.

$$\sum_{p=\max[1, t-m_p+1]}^t C_{pst} (\omega_{p,2(s-1)} + \omega_{p,2s-1}) \begin{cases} \geq \\ = \\ \leq \end{cases} C_t \quad \forall t \quad (3.9)$$

$$t = 1, \dots, n$$

where

C_t is the bound payments that may be made during model period t , associated with introductions during or prior to period t .

Note that, for the introduction case I2, the system (3.9) provides a piece-wise linear approximation to payments incurred, whereas it is exact in cases I1 and I3.

3.3.1.4 The Fixed Charge Problem

The budgeting mechanism developed in subsection 3.3.1.3 dealt with ensuring that capacity introductions were financially feasible: that is, that repayments required to be made to offset the cost of introducing capacity were within the prescribed bounds specified by system (3.9). This subsection deals with incorporating cost measures associated with introduction into an economic efficiency measure (an objective function). These costs which for a given facility are independent of the level of throughput, are fixed costs. The problem posed by these costs within a mathematical program is a fixed charge problem and is a standard problem in Operations Research (Hadley, 1964; Gass, 1975; Taha, 1976).

Using the elements from the set S_1 , the system (3.1), and the sets of sequences (3.8) representing introduction payments, it is an easy matter to develop a fixed-charge mechanism that provides a piece-wise linear approximation

to the present value at the base year of the model (usually taken to be the start of the first model period) of costs required to introduce facilities during the planning horizon. This is expression (3.10), which is non-positive in that all terms except C'_{pt} are positive.

$$\sum_p \sum_{t=1}^n C'_{pt} (\omega_{p,2(t-1)} + \omega_{p,2t-1}) \quad (3.10)$$

$$C'_{pt} = -(1+\alpha)^{-(t-1)} \sum_{q=t}^{\min[n, t+m_p-1]} C_{ptq} (1+\alpha)^{t-q}$$

where

- α is the discount rate associated with introduction payments,
- C'_{pt} is the negative of the present value at the commencement of the planning horizon of payments incurred during the planning horizon in order to introduce processing facility p at the start of period t .

Note that all the mechanisms proposed in this study for the budgeting and fixed-charge problems concern themselves only with payments that are made during the planning horizon. This approach has been adopted because it avoids situations of having to make valuations at the end of the planning horizon, which could arise if payments associated with introductions are incomplete.

Consideration of asset valuation is deliberately omitted within an economic efficiency measure because current methods of valuing are imprecise and may introduce stochastic elements within an efficiency measure because of their departure from an ill-defined actual value. Leslie (pers. comm., 1984) considers good valuations to be those that are consistent, in the sense that they can be repeated by different valuers without wide variation. Accuracy in terms of closeness to actual value assumes a secondary role because actual value is

usually ill-defined. Moreover, it may be better to consider the earning power of assets alone in an efficiency measure this being likely to be more precisely quantifiable than measures employing both earning power and valuation of assets.

3.3.2 Non-convex Non-interpolating Binary Mechanism

The introduction and regulation mechanism developed in this subsection is non-convex and thus requires the use of integer programming techniques to resolve the non-convex programming problem. However, it differs from the mechanism developed in subsection 3.3.1 in that it does not employ continuous weights (namely, $\omega_{2t-1}, \omega_{2t}$; $t=1, \dots, n$ of S_1). Consequently, introductions must now be considered to occur at a single time point within a model period. In this study, this point is arbitrarily chosen to be the start of the model period. The development of the mechanism in this subsection entirely parallels that given in subsection 3.3.1, and parallel constraint systems and expressions are indicated. Once again, the development is largely with respect to a single facility at a specified location.

Let

$$S_2 = \{y_t : y_t = 0 \text{ or } 1; \quad t = 1, \dots, n\} \quad (3.11)$$

where

$$y_t = \begin{cases} 1 & \text{introduction of facility at start of model period } t, \\ 0 & \text{otherwise.} \end{cases}$$

Once again, the location problem requires that there be an introductory regulatory mechanism, for each facility, at each location, where capacity introductions may occur. This requires reinstatement of the subscript p , denoting location and type of processing facility (see subsection 3.3.1.1).

The resource allocation problem is exactly as specified in subsection 3.3.1.2. Capacity utilisation necessitates consuming input resources and producing output resources, whereas the non-availability of capacity is associated with non-operation of processes. Thus, (3.12) and (3.13) parallel (3.4) and (3.5) respectively for the non-interpolating mechanism. All terms are as previously defined. Again, it is a minor problem to formulate the system for resources that are consumed.

$$\sum_{s=1}^t r_{is} y_s - \sum_{k \in K_i} a_{ki} x_{tk} \leq 0 \quad \forall i, t \quad (3.12)$$

$$\sum_{s=1}^t r_{is}^* y_s - \sum_{k \in K_i} a_{ki} x_{tk} \geq 0 \quad \forall i, t \quad (3.13)$$

$$i = 1, \dots, I$$

$$t = 1, \dots, n$$

The systems (3.11), (3.12) and (3.13) allow introductions to occur only at a fixed point in a model period, in this study this point is taken to be the start of the model period. Similarly, by reinstating the subscript p , systems can be developed to allocate resources to and from transportation arcs that start and finish respectively at the specified location. These are the constraints (3.6) and (3.7).

The budgeting problem is as stated in subsection 3.3.1.3. However, for the introductory regulatory mechanism under consideration there are no continuous weights to be used in its resolution. Instead, direct use must be made of the binary variables from the set S_2 . Thus system (3.14) parallels the system (3.9).

$$\sum_p \sum_{s=\max[1, t-m_p+1]}^t C_{pst} y_{ps} \left\{ \begin{array}{l} \geq \\ = \\ \leq \end{array} \right\} C_t \quad \forall t \quad (3.14)$$

$$t = 1, \dots, n$$

where

y_{ps} is the binary variable associated with the introduction of processing facility p in model period s .

The fixed-charge problem is as stated in subsection 3.3.1.4. However, only binary variables from S_2 are used in its resolution. Thus to parallel (3.10) is (3.15). This expression is again non-positive because introduction payments represent a cost (loss) to the modelled system. All terms are as previously defined.

$$\sum_p \sum_{t=1}^n C'_{pt} y_{pt} \quad (3.15)$$

3.3.3.1 Constant Returns to Scale

The introductory regulatory mechanism developed in this subsection parallels the non-interpolating binary mechanism of subsection 3.3.2. This mechanism is found to be the most readily applicable of the convex mechanisms developed in this study (c.f. subsection 3.3.3.2).

Let

x_t be the proportion of the processing facility that is introduced at the start of time period t .

Then, the inequalities (3.16) parallel the system (3.11)

$$0 \leq x_t \leq 1, \quad \forall t \quad (3.16)$$

$$t = 1, \dots, n.$$

The location problem is treated as in subsection 3.3.2, and requires the use of the subscript p , denoting location and type of facility. This will be reinstated as required.

For the resource allocation problem (see subsection 3.3.1.2), only upper bounds are permissible because the presence of lower bounds would create a non-convexity. Thus, the inequalities (3.17) parallel (3.13) in that each inequality represents an upper bound on the consumption of resource i in period t .

$$\sum_{s=1}^t r_{is}^* x_s - \sum_{k \in K_i} a_{ki} x_{tk} \geq 0 \quad \forall i, t \quad (3.17)$$

$$i = 1, \dots, I$$

$$t = 1, \dots, n.$$

The budgeting problem (see subsection 3.3.1.3) can be expressed in terms of the system (3.16) provided only upper bounds are used on payments associated with introductions. This restriction is necessary in order to generate a convex constraint set. The system (3.18) then parallels (3.14).

$$\sum_p \sum_{s=\max[1, t-m_p+1]}^t C_{pst} x_{ps} \leq C_t, \quad \forall t \quad (3.18)$$

The fixed-charge problem (see subsection 3.3.1.4) strictly no longer exists in that capacity is now regarded as being continuous between its lower bound zero and upper bound imposed by the system (3.17). The fixed-charge problem exists only in a continuous analogue form in this situation. This is (3.19), the terms of which are included in an economic efficiency measure and are always strictly non-positive for the reasons outlined in subsection 3.3.2.

$$\sum_p \sum_{t=1}^n C'_{pt} x_{pt} \quad (3.19)$$

Continuous introduction mechanisms, as presented in the systems (3.16) through (3.19), are attractive in that all the

required variables are continuous and enter the model in a linear fashion. Thus, if these mechanisms are included in programs that are otherwise linear, then clearly this obviates the need to solve programs with integer variables. However, such an introduction mechanism is weak in so far as it permits fractional processing facilities to be introduced. A major reason why the non-convex mechanisms of subsections 3.3.1 and 3.3.2 are presented in this study is that they avoid situations like this that may be unrealistic as the size of processing facilities increases.

3.3.3.2 Diminishing Returns to Scale

The introductory regulatory mechanism developed in this subsection consider introductions when there are diminishing returns to scale with respect to utilisation of capacity (that is, increasing marginal costs for capacity units). To ensure the mechanism is convex, capacity is regarded as being infinitely divisible. It will be seen that this approach is of little utility as an "introduction mechanism" because of the difficulty associated with the analogue of the fixed-charge problem.

Assume that capacity at a facility may be defined in terms of the production or consumption of a single resource. Should the capacity of a facility be increased, moreover, then the cost of additional capacity follows a relationship such as that given in Figure 3.3. Thus, it becomes progressively more expensive to increment capacity at a plant. The first r^* units have introduction cost per unit given by the slope of the linear approximation between 0 and r^* , while those for the second r^* units are given by the slope of the linear approximation between r^* and $2r^*$, and so on.

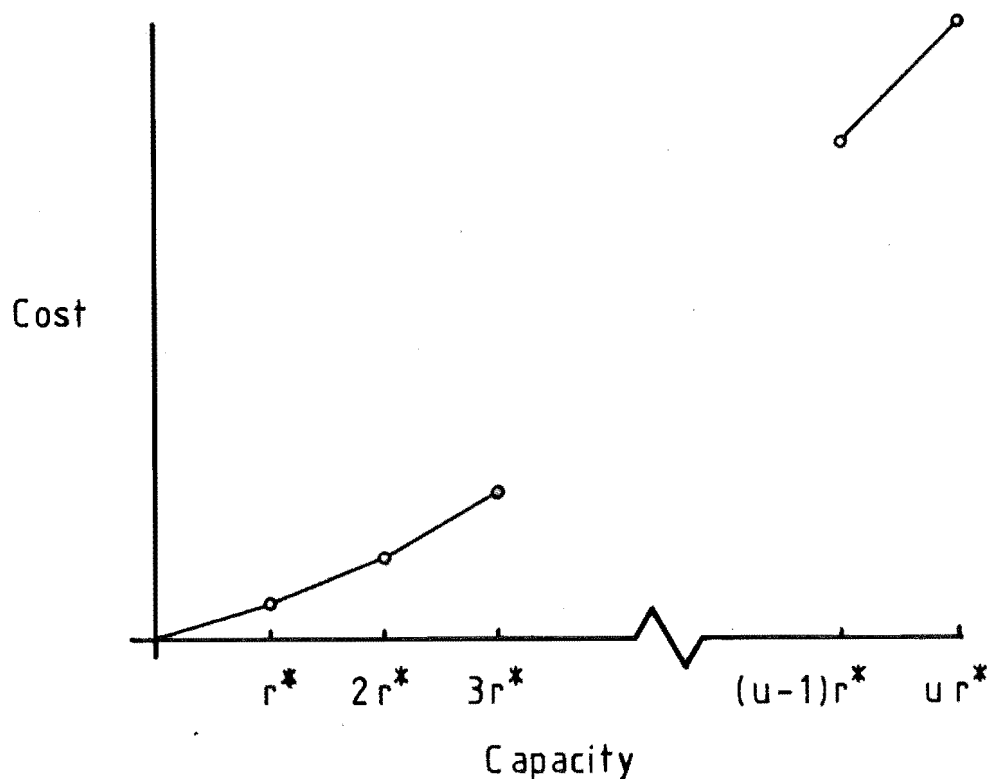


Figure 3.3 : Introduction Cost vs Capacity

Shown is the functional relationship between introduction costs and capacity, for a single facility with diminishing returns to scale with respect to utilisation of capacity.

A requirement for a valid introduction that allows for possible subsequent additions to capacity is that capacity units be utilised in the correct order. That is, utilisation of capacity must start with the first r^* capacity units and then the next r^* units, and so forth (that is, from left to right in Figure 3.3).

The following definitions are required for the development of the introductory regulatory mechanism.

Let

- x_{ut} be the proportion of capacity increment u that is taken up at the start of period t , and
- r^* be the size of the capacity increment measured in units of a single resource either produced or consumed during a model period.

The inequalities (3.20) then parallel (3.11). Their function however, is markedly different in that each capacity increment in (3.20) can comprise portions that are added in each period.

$$\sum_{t=1}^n x_{ut} \leq 1 \quad \forall u \quad (3.20)$$

$$u = 1, \dots, U.$$

Once again, the resource allocation problem permits only upper bounding constraints because the presence of positive lower bounding constraints other than zero would be a non-convexity or would imply that the facility was already in operation. Thus, the inequalities (3.21) that impose upper bounds on resources produced parallel (3.13).

$$\sum_{s=1}^t \sum_{u=1}^U r_i^* x_{us} - \sum_{k \in K_i} a_{ki} x_{tk} \geq 0 \quad \forall i, t \quad (3.21)$$

where

r_i^* is the upper limit for the production of resource i , when r^* capacity units are available ($r_i^* \leq r^*$), for a specified facility.

The problem with this formulation is to ensure that the capacity is utilised in the correct order. Consequently, certain relationships amongst coefficients of an objective function must hold. In discussing the fixed charge problem for this introduction mechanism, indications are given as to why these relationships may not hold.

The budgeting problem permits only constraints that give rise to a convex set. For simplicity of subscripting, the development is given for a single processing facility p as (3.22).

$$\sum_{u=1}^u \sum_{s=\max[1, t-m+1]}^t C_{ust} x_{us} \leq C_t \quad (3.22)$$

$$t = 1, \dots, n$$

where

C_{ust} is the payment made in period t to introduce a capacity increment u at the start of period s .

u is the number of capacity increments available, each of size r^* units.

m is the number of model periods over which payments are made in order to introduce r^* capacity units.

all other terms are as previously defined.

Once again, as in the constant returns to scale case, the fixed-charge problem strictly no longer exists and only has a continuous analogue. However, the introduction costs must be included within the economic efficiency measure, this being a necessary requirement to ensure that capacity is utilised in the correct order. For a single facility p , the resulting expression is (3.23).

$$\sum_{t=1}^n \sum_{u=1}^u C_{ut} x_{ut} \quad (3.23)$$

where

$$C_{ut} = -(1+\alpha)^{-(t-1)} \sum_{q=t}^{\min[n, t+m-1]} C_{utq} (1+\alpha)^{t-q}$$

C_{ut} is the negative of the net present value at the beginning of the planning horizon of payments made during the planning horizon in order to introduce the capacity increment u at the start of period t .

The requirement that capacity be utilised in the correct order is that, utilising the first capacity increment is always cheaper than utilising the second, and so forth. This can be expressed in terms of the coefficients C_{ut} in (3.23) as condition (3.24).

$$\left\{ C_{\ell m} < C_{pq} : \text{for } \ell < p; \ell, p \in \{1, \dots, u\}; m, q \in \{1, \dots, n\} \right\} \quad (3.24)$$

Condition (3.24) is an ordering relation amongst sets of real numbers (C_{ut} 's) computed using a discount rate. It may be difficult to satisfy this requirement in practice, and so continuous introduction mechanisms with increasing costs for capacity utilisation may be of little practical use. Seemingly, the more simple situation of constant returns to scale offers greater scope for implementing convex introduction mechanisms than does diminishing returns to scale.

3.4 MARKETING MECHANISMS

The simplest and most often formulated marketing mechanism used in linear programs or mixed integer linear programs assumes perfect competition in products markets. In a purely competitive market, producers are price takers and are unable to influence price by the quantity produced. Here, market sales illustrate constant returns to scale (Boulding and Spivey, 1960).

Another common marketing mechanism is to assume diminishing returns to scale in products markets, where the amount that can be sold at each price is known initially and constitutes part of the information defining the market structure. Linear programs, in this situation, allocate products for sale at the highest available market price that has unused capacity associated with it.

Modelling increased returns to scale cannot be done with linear programming, *per se*, because this gives rise to a non-convex programming problem (Williams, 1978). It is

possible using Separable Programming Techniques to model economies of scale (LP with modified basis entry), but such programs no longer satisfy the conditions necessary to locate a global optimum, and a local optimum may result. Mixed integer linear programming techniques can be used to model economies of scale. However, application of this for a FUMP would normally be prohibitive in terms of the number of integer variables required.

Thus, if continuous variables are used to formulate marketing mechanisms with linear constraints and objective functions, the mechanism must involve constant and/or diminishing returns to scale. Additionally, capacity constraints may be imposed to bound and/or smooth material sold. But even this is lamentably short of the sophistication necessary to deal with these important complex mechanisms.

The importance of marketing mechanisms is directly related to their function as revenue generators. Thus, economic viability in the long term depends on what happens there. Leslie (cited in Richardson, 1985) illustrates this well.

"Forest sector planning, either for or in the sector must start from the export market outlooks just as forest management must be directed by them. As an issue in the general issue of forest sector planning assessment of export market prospects has no equal importance."

Their complexity, on the other hand, is linked to their highly stochastic nature. This means that the definition of market structures by location, price, quantity, and quality for each product over time becomes a statistical problem. Knowledge of these structures will at best be

limited because the theory of statistics indicates that any estimates made in a stochastic environment will have errors of estimation attached.

The importance and complexity lead to an examination of the feasibility of applying stochastic programming techniques to market structures. The essential idea of stochastic programming techniques is to convert a probabilistic problem into an equivalent deterministic problem (Hillier and Lieberman, 1974; Taha, 1976)³. Stochastic programming techniques are classified as either stochastic or chance-constrained. In brief, stochastic models require the solution to be feasible for all combinations of model parameters used to define the problem, whereas chance-constrained programming permits feasible solutions to violate those constraints that are chance-constrained with a small probability.

Both stochastic and chance-constrained programming techniques could be applied to marketing mechanisms, but stochastic programming offers the most readily identifiable advantage because it does not require the estimation of variances or covariances and is easier to implement. Only stochastic programming is therefore considered further in this study. The envisaged application depends on the detailing of several market structures where, previously, only one was described. Each new structure is assumed to be possible with a specified probability of occurrence. Then, the associated linear program (deterministic representation) could be formulated according to the procedure outlined in Wagner (1975). Such a model is called a two-stage stochastic

³ Stochastic programming techniques refer to decision-making under risk, where it is assumed the distributions of random variables are known.

model, the solution of which would allow an examination of the effects of a varying market.

To illustrate a possible example of a two-stage stochastic procedure, consider an uncertainty over future prices in a products market where all produce can be sold, all other parameters being assumed deterministic. Scenarios of high, medium and low market prices could be constructed, each of which has an estimated probability of outcome. With stochastic programming, these markets could be represented jointly in a model, thus in effect modelling a market with varying market prices. Such a procedure could be used to search for a develop "robust strategies". That is, a system is considered to be robust if it performs well under a wide range of conditions, possibly by dynamically altering its response to its environment. Such robust strategies may be characterised in terms of both stability and economic efficiency. The idea of robustness is not easy to formulate explicitly as part of an optimising problem, the difficulty being that robustness is such an ill-defined notion, whose meaning is open to interpretation. Despite this difficulty, the notion is still important since robust strategies may perform adequately economically but not have the same risk associated with them as would non-robust strategies that purport to perform better economically.

3.5 STRUCTURAL CONSIDERATIONS FOR FOREST UTILISATION MANAGEMENT PROBLEMS

This section discusses solution techniques for FUMPs that give recognition to the structural aspects of the problem. In subsection 3.5.1, a special partitioning procedure that may be applied to FUMPs that includes non-convex introduction mechanisms is discussed. In subsection 3.5.2, FUMPs are considered from the viewpoint of specialised network problems, and are shown to be consistent with a processing network when convex introduction mechanisms are employed.

3.5.1 Benders Algorithm

Forest Utilisation Management Problems that include non-convex introduction mechanisms may be represented by the class of program P1. The partitioning procedure developed by Benders (1962) may be used to solve problems of this form.

$$\begin{array}{ll}
 \min & cx + dy \\
 \text{s.t.} & Ax + Fy \geq b \\
 & x \geq 0, \quad y \text{ binary}
 \end{array}
 \qquad \text{P1}$$

The derivation of Bender's algorithm requires Farkas' lemma and statements concerning polyhedral cones. These topics are covered in the texts of Hadley (1961, 1964). Bender's procedure separates the set of variables into a continuous set associated with the matrix A and a binary set associated with the matrix F. This separation is shown by Lasdon (1970) to lead to P2, an equivalent problem to P1.

min z

s.t.

$$z \geq dy + (b - Fy)u_i^P, \quad i=1, \dots, n^P \quad P2$$

$$(b - Fy)u_i^r \leq 0, \quad i = 1, \dots, n^r$$

where

u_i^P is an extreme part of $\{u | A^T u \leq c, u \geq 0\}$,

u_i^r is an extreme ray of $\{u | A^T u \leq c, u \geq 0\}$, and

$n^P(n^r)$ is the number of extreme points (extreme rays) of $\{u | A^T u \leq c, u \geq 0\}$

Lasdon (*op. cit.*) shows that the equivalence of P1 and P2 means that, if (z^0, y^0) solves P2, then (x^0, y^0) solves P1, where x^0 is the value for x obtained by solving P1 with y fixed.

Because the constraints of P2 are not available initially, and constraint numbers may be very large (they are obtained from extreme points and extreme rays of a convex set), problem P2 is solved by a relaxation technique, constraints being added at successive iterations if a violation of the constraint set of P2 is detected. When the algorithm terminates, then either problem is infeasible, the problem is unbounded, or an optimal solution has been obtained.

Step 2 of Bender's procedure (see Lasdon, *op. cit.*) may be formulated as a pure-integer programming problem. As such, a large number of procedures could be used to solve it (Balinski, 1965). However, Balas (1967) has designed a special-purpose enumerative technique to deal with the integer programs that arise during Bender's procedure and is obviously preferable in this instance.

Step 3 requires the determination of a solution to problem P3, which is usually accomplished by solution

of the dual problem. Solution of either of the problems expressing P3 involves solving a linear program with coupling constraints, coupling variables, and block diagonal structure. Ritter (1967) has developed a partitioning procedure that can be applied to such programs.

Primal P3⁴

$$\begin{aligned}
 &\min c_0 x_0 + c_1 x_1 + c_2 x_2 + \dots + c_p x_p \\
 &\text{s.t.} \\
 &\quad D_0 x_0 + A_1 x_1 + A_2 x_2 + \dots + A_p x_p = b_0 - F_0 y \\
 &\quad D_1 x_0 + B_1 x_1 = b_1 - F_1 y \\
 &\quad D_2 x_0 + B_2 x_2 = b_2 - F_2 y \\
 &\quad D_p x_0 + B_p x_p = b_p - F_p y \\
 &\quad x_0 \geq 0, x_1 \geq 0, x_2 \geq 0, \dots, x_p \geq 0
 \end{aligned}$$

Dual P3

$$\begin{aligned}
 &\max (b_0 - F_0 y) u_0 + (b_1 - F_1 y) u_1 + \dots + (b_p - F_p y) u_p \\
 &\text{s.t.} \\
 &\quad D_0^T u_0 + D_1^T u_1 + \dots + D_p^T u_p \geq c_0 \\
 &\quad A_1^T u_0 + B_1^T u_1 \geq c_1 \\
 &\quad A_2^T u_0 + B_2^T u_2 \geq c_2 \\
 &\quad A_p^T u_0 + B_p^T u_p \geq c_p
 \end{aligned}$$

Where the constraint matrix for the primal form is A and that for the dual is A^T , the vector b has been partitioned as $[b_0 | b_1 | \dots | b_p]^T$. Similarly, the rows of F have been partitioned as F_0, F_1, \dots, F_p where F_0 is the submatrix formed by the first $|b_0|$ rows of F , F_1 is the submatrix formed by the next $|b_1|$ rows of F , and so on.

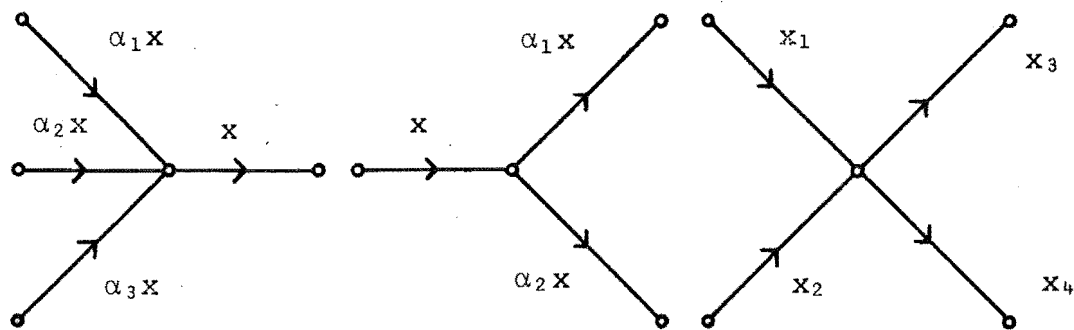
⁴ The structure of the primal problem in P3 arises in the following manner. The coupling variables represent activities associated with the Forest Management Problem and any weights employed in non-convex introduction mechanisms. The coupling constraints represent limitations on such jointly shared resources such as labour and capital. The block diagonal elements represent the transportation, processing, and marketing activities of a single model period.

This application of Bender's procedure preserves the structure of A in $P1$, where it may be taken into account at step 3. Moreover, a specialised integer programming algorithm exists for the programs that arise at step 2. The Branch and Bound method for solving $P1$ has a comparative disadvantage as the size of the problem increases in that it requires both large amounts of computer storage (details of problems solved and problems remaining to be solved are stored within a tree structure, which can become very large) and computer time (branch and bound procedures solve LPs at each iteration, but the number of LPs to solve invariably becomes large as the number of integer activities is increased). Bender's procedure does not share the first disadvantage of storage requirements, and it may not share the second regarding execution times.

3.5.2 Processing Networks

When seeking solution techniques that give consideration to structural aspects of Forest Utilisation Management Problems, an alternative avenue is to consider network formulations to such problems. Situations involving production alternatives can be usually formulated as network problems, where conservation of flow in both nodes and arcs is ensured, and where capacity bounds for arc flows exist. Additionally, side activities and/or side constraints may be needed to specify processes (Bazaraa and Jarvis, 1977; Koene, 1983). Network problems such as these may be represented by a class of network known as processing networks basically comprised of three types of process: blending, refining, and transportation, examples of which are shown in Figure 3.4.

Figure 3.4 : Process Types in a Processing Network



$$\sum_i \alpha_i = 1, \alpha_i > 0, \forall i \quad \sum_i \alpha_i = 1, \alpha_i > 0, \forall i$$

(a)

(b)

(c)

Shown above are:

- (a) A *blending process* allowing the possibility of several components to be blended in given proportions;
- (b) A *refining process* allowing a flow to be split into several components in given proportions;
- (c) A *transportation process* allowing transportation of material between nodes, no assumptions are made concerning proportionality of flow.

A Forest Utilisation Management Problem consists largely of production processes and transportation activities that are integrated through the use of resources. Thus a processing network with additional side constraints may be used to represent these problems. If introductions are to be considered for processing facilities, then the convex introduction mechanisms discussed in subsection 3.3.3 could be adapted for use as part of a processing network. The most applicable of these was identified as the constant returns to scale case. Non-convex introduction mechanisms such as those within subsections 3.3.1 and 3.3.2 could be

considered by adaptation to a processing network form and embedding the network problem within a branch and bound procedure.

This latter approach offers an attractive alternative to branch and bound procedures using the simplex method with simple upper bounds because network algorithms generally show computational advantage over traditional simplex based techniques (Koene, *op. cit.*). Moreover, as indicated in the next paragraph processing networks can be used to solve Forest Management Problems.

A recent study by Garcia (1984) has proposed the use of pure network with additional side constraints to solve Forest Management Problems. Area regulation is consistent with flow through a pure network. Thus, the structural constraints (2.12) through (2.15) of subsection 2.2.4 specify the pure network. The regulatory constraints discussed in subsection 2.2.5 and section 2.3 constitute the additional side constraints. Problems formulated in this way can be regarded as either a processing network, with only transportation nodes and additional side constraints, or else as an LP with an embedded pure network structure. Using the first interpretation, they can be solved using the Simplex PRON procedure of Koene (*op. cit.*), while for the second interpretation, they can be solved using the Simplex SON procedure of Glover and Klingman (1981), as suggested by Garcia (*op. cit.*).

CHAPTER 4

SPATIAL AND TEMPORAL EFFECTS OF ROUNDWOOD AVAILABILITY

This chapter addresses the problem of determining what may be produced by a processing facility that receives a restricted set of inputs. The processing activities at a facility are assumed to be represented as a multi-stage process. The set of inputs to the facility could be anticipated to vary in response to temporal or spatial effects. For example, classes of roundwood utilisable by the facility do not become available because stands in forests supplying the facility are not yet old enough to produce such classes, or classes of roundwood do not become available because no feasible transportation links exist between the site of the facility and forests producing those classes of roundwood.

This problem is important in that it could possibly prevent redundancy in problems where the utilisation of roundwood is considered. Consider, for example, a facility with a multi-stage production process that cannot possibly produce a specified production set output from the final production stage, and yet the production of this set is considered necessary for the operation of the facility. This then suggests that the facility should not be included as part of a multi-period planning problem during the periods it is not capable of producing all elements in the specified production set.

Similarly, another criterion that could be used to decide whether to exclude facilities, as part of a multi-

period planning problem, is as follows. Exclude the facility when it cannot produce a specified production set, in a set of adjoining model periods, inclusive of the final model period.

Generally, model construction (matrix generation for FUMPs) can be undertaken in a manner that does not allow redundancy to occur. This is possible because information available during construction can be used to detect and avoid possible redundancies. This chapter is concerned with redundancies that may occur at facilities having a multi-stage production process. These are a particularly important form of redundancy for facilities whose capacity is regulated using binary variables because it may allow binary variables to be excluded from the problem. For example, if during model construction, redundancy of a facility is detected at a model period, then the continuous variables representing the facility along with the regulating integer variable may be excluded from the problem for the model period specified.

The mechanisms developed in this chapter for detecting redundancies in multi-stage processes would be particularly amenable to network formulations of Forest Utilisation Management Problems in that a digraph involved in presenting a multi-stage manufacturing process, termed a "manufacturing digraph" is closely related in form to a processing network described in subsection 3.5.2. For FUMPs specified as linear or mixed integer linear programs, redundancies are normally removed by application of reduction theorems to the generated problem. These are methods by which redundant equations, null variables, and non-external variables can be removed from the problem to create a similar system of

linear equations (Luenberger, 1973). This reduction can be automated and is usually a standard feature of mathematical programming systems (Brearley, *et al.*, 1975; cited in Williams, 1978).

Section 4.1 introduces a problem in terms of a "resource digraph" and presents solution mechanisms involving backward paths and forward and backward mappings. Section 4.2 details a solution technique in relation to a digraph closely related to the resource digraph, the manufacturing digraph. Section 4.3 details a technique derived from consideration of the forward and backward mappings. Section 4.4 suggests how such theory may be implemented during an FUMP generation phase.

4.1 THE RESOURCE DIGRAPH APPROACH

A digraph can be used to indicate resource flows within a multi-stage production process. This digraph represents inputs to the initial stage, output from the final stage, and outputs and inputs from intermediate stages and shall, in this study, be termed a *resource digraph*. Using such a digraph, Robinson (see Appendix 4.1) has formulated a solution to the problem of determining the largest possible set of products that can be output from the final stage of a multi-stage production process when a subset of the resources used as input to the first stage is available. Note that this solution, along with all other solutions in this chapter, is based on information gained from the presence or absence of first stage input resources, nothing is indicated about resource levels or proportions in which resources are combined during processing.

A solution procedure involving the construction of backward paths is initially illustrated in Example 4.1, then the problem is formalised, with the structure of the resource digraph being introduced along with the solution procedure formulated by Robinson (*op. cit.*). This latter solution procedure is very simple, making the possible automation of the procedure a tractable exercise.

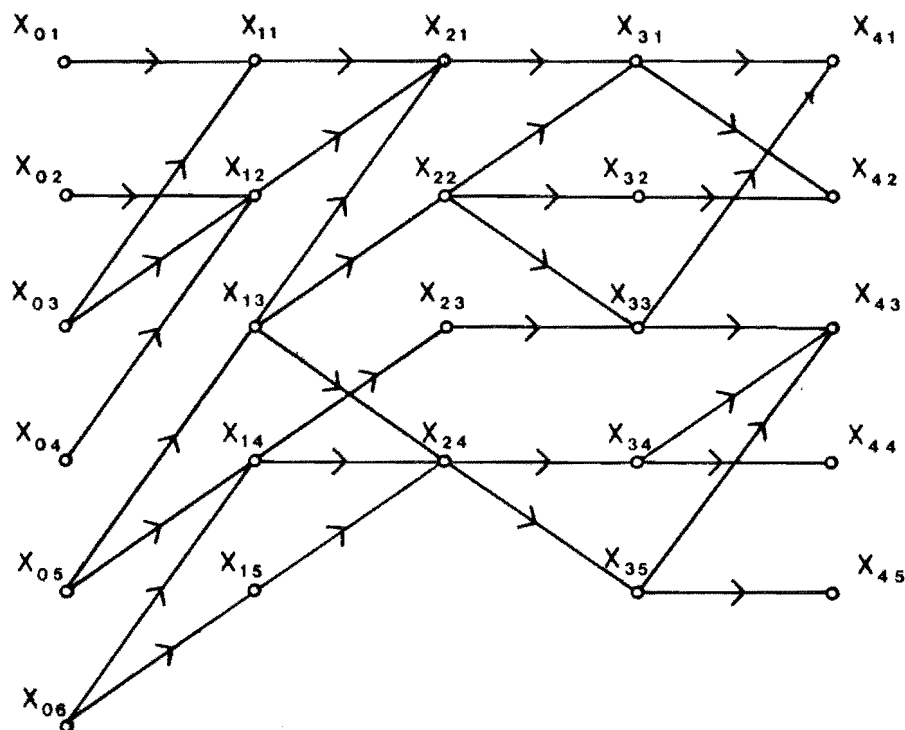


Figure 4.1 : Resource digraph for a four-stage process

where

vertices labelled x_{ij} denote resource element j , either used as input to stage $i+1$, or output from stage i , or both. Output of any resource element x_{ij} , $i \geq 1$, $j = 1, \dots, n_i$ is contingent upon the availability of all resource elements in stages 0 through $i-1$ that are encountered by the traversal of any path emanating at x_{ij} and ending at an element from $\{x_{0j}\}$, these paths are formed by the backward traversal of arcs. e.g. x_{12} requires resource elements $\{x_{02}, x_{03}, x_{04}\}$; x_{21} requires the first stage set $\{x_{11}, x_{12}, x_{13}\}$ and the zero stage set $\{x_{01}, x_{02}, x_{03}, x_{04}, x_{05}\}$ since all these vertices lie on backward paths from x_{21} .

Example 4.1

Given the resource digraph in Figure 4.1, the problem is to determine the largest set possibly produced by the final production stage when a set of initial resources

$U_0 = \{x_{04}, x_{05}, x_{06}\}$ is available, and further find which elements of U_0 are required to produce this set.

The resource set $U_0 = \{x_{04}, x_{05}, x_{06}\}$ may be used to produce elements from $\{x_{13}, x_{14}, x_{15}\}$, output from the first production stage. It is not possible to produce elements from $\{x_{11}, x_{12}\}$ because backward paths from these elements end in points that are not contained in set U_0 . Denote the set $\{x_{13}, x_{14}, x_{15}\}$ by ϕU_0 (the set-naming convention adopted in this example will be explained later). Elements from this set may be introduced as input to the second stage process; then backward paths from elements output by the second production stage must include an element from the set ϕU_0 and an element from the set U_0 , these being the only preceding elements available for utilisation. This enables the elements $\{x_{22}, x_{23}, x_{24}\}$ to be produced at the second production stage. Denote this set by $\phi\phi U_0$. Continue this approach, that an element can be produced at a stage only if backward paths include an element from each of the previous stage sets that can be made available. The third and fourth production stage enable the sets $\{x_{32}, x_{33}, x_{34}, x_{35}\}$ and $\{x_{43}, x_{44}, x_{45}\}$ to be produced, denote these sets by $\phi\phi\phi U_0$ and $\phi\phi\phi\phi U_0$ respectively.

To consider what initial resources are required in order to produce the elements $\{x_{43}, x_{44}, x_{45}\}$, the backward paths of elements in this set are examined. Because no backward path includes x_{32} , which is capable of being produced, only $\{x_{33}, x_{34}, x_{35}\}$ is required as input to the fourth production stage. Denote this set by $\beta\phi\phi\phi\phi U_0$. This set is sufficient to produce a final stage output set. Thus, only backward paths from elements in this set need be considered in order to produce the final stage output set.

These backward paths require the set $\{x_{22}, x_{23}, x_{24}\}$ be available as input to the third production stage. Denote this set by $\beta\beta\phi\phi\phi U_0$. Continuing in the manner, the sets $\{x_{13}, x_{14}, x_{15}\}$ and $\{x_{05}, x_{06}\}$ are required as input for the second and first production stages. Denote these sets by $\beta\beta\phi\phi\phi\phi U_0$ and $\beta\beta\beta\phi\phi\phi\phi U_0$ respectively. The set $\{x_{05}, x_{06}\}$ is the subset of U_0 that is required in order to produce the largest possible set output from the final production stage. \aleph

Although example 4.1 shows a method of gaining a solution to the problem under consideration, it requires the examination of elements (vertex labels) on backward paths. This is a complexity that can be avoided by use of appropriately defined mapping procedures between subsets of the vertices in the resource digraph. This is the solution procedure established by Robinson (see Appendix 4.1), the discussion of which requires the introduction of the structure of the resource digraph and definition of the forward and backward mapping procedures ϕ and β respectively.

The resource digraph $D(V,A)$ has vertex set V partitioned into $n+1$ subsets V_0, V_1, \dots, V_n . These represent the resource elements at various production stages. Each arc of D joins a vertex in the set V_i to a vertex in the set V_{i+1} . The set V_0 , consisting of sources, is the set of resources for the first production stage, the set V_n , consisting of sinks, is the set of resources from the final production stage (Robinson and Foulds, 1980; Wilson, 1979).

The forward map $\phi: \mathcal{P}(V_i) \mapsto \mathcal{P}(V_{i+1})$ maps subsets of V_i to subsets of V_{i+1} for $0 \leq i \leq n-1$. For $X \subseteq V_i$, the rule (4.1) defines ϕ .¹

¹ The notation $\mathcal{P}(A)$ is used to denote the power set of A , that is, the set of all possible subsets of A .

$$\phi(X) = \{v : v \in V_{i+1}, \text{ all precursors of } v \text{ are in } X\} \quad (4.1)$$

Thus, $\phi(X)$ is the image of X under ϕ and is the set of vertices, perhaps with common labels, capable of being produced when given X , itself a vertex set with possibly common labels. This distinction between vertices and vertex labels allows the construction of the resource digraph when alternative ways exist to form resources. The set $\phi(X)$ always consists of distinct vertices. Additionally it will correspond to a distinct set of vertex labels when there exists only one way to produce each vertex contained in this set.

The backward map $\beta: P(V) \rightarrow P(V_{i-1})$ maps subsets of V_i to subsets of V_{i-1} for $1 \leq i \leq n$. For $X \subseteq V_i$ the rule (4.2) defines β .

$$\beta(X) = \{u: u \in V_{i-1}, \text{ at least one successor of } u \text{ is in } X\} \quad (4.2)$$

Thus $\beta(X)$ is the image of X under β and is the set of vertices, perhaps with common labels, required to produce X , itself a vertex set with possibly common labels. The labelling of vertices is further explored in section 4.2 where repetition of vertex labels (note - not repetition of vertices) is used in the construction of resource digraphs when resources can be formed in more than one way.

Given a set of initial resources $U_0 \subseteq V_0$ then, by using the maps ϕ and β , it is possible to identify subsets X_0, X_1, \dots, X_n of V_0, V_1, \dots, V_n respectively such that the following are satisfied.

- (i) $X_0 \subseteq U_0$
- (ii) If $x_i \in X_i$ for $1 \leq i \leq n$,
then all precursors of x_i are in X_{i-1} .

(iii) If $x_i \in X_i$ for $1 \leq i \leq n-1$,

then at least one successor of x_i is in X_{i+1} .

(iv) X_n is the largest possible².

The subsets X_0, X_1, \dots, X_n are chosen as follows:

$$X_0 = \beta^n \phi^n U_0, \quad X_1 = \beta^{n-1} \phi^n U_0, \dots, X_i = \beta^{n-i} \phi^n U_0, \dots, X_n = \phi^n U_0$$

Each of these subsets is the image of a composite map mapping $U_0 \subseteq V_0$ to $X_i \subseteq V_i$ for $i=0, \dots, n$. These sets solve the problem of identifying what needs to be produced at each stage in order to generate the largest possible resource set from the final production stage. The following interpretation is made of the composite map.

$$\begin{aligned} x &= \beta^{n-i} \phi^n U_0 \\ &= \underbrace{\beta(\beta(\dots \beta}_{n-i \text{ terms}}(\underbrace{\phi(\phi(\dots \phi}_{n \text{ terms}}(U_0) \dots))) \dots))}_{n-i \text{ terms}} \end{aligned} \quad (4.3)$$

The sets X_0, X_1, \dots, X_n are formed in reverse order. That is, $X_n = \phi^n U_0$ is formed first, then $X_{n-1} = \beta \phi^{n-1} U_0$, and so forth until $X_0 = \beta^n \phi^n U_0$ is formed. This mapping procedure is illustrated in Table 4.1, where example 4.1 is solved using the mapping approach. The set nomenclature adopted in example 4.1 actually constitutes the composite maps, excepting U_0 and ϕU_0 , required to generate the respective sets, for example $\phi \phi U_0 = \phi(\phi(U_0)) = \{x_{22}, x_{23}, x_{24}\}$.

² For ease of exposition in stating (i) through (iv), a resource element has been denoted by x_i , strictly they should be denoted by x_{ij} to be consistent with the levelling technique used in examples in this chapter.

Table 4.1 : The Forward-Backward Mapping Procedure

U_0	ϕ	ϕ^2	ϕ^3	ϕ^4	$\beta\phi^4$	$\beta^2\phi^4$	$\beta^3\phi^4$	$\beta^4\phi^4$
X_{04}	X_{13}	X_{22}	X_{32}	X_{43}	X_{33}	X_{22}	X_{13}	X_{05}
X_{05}	X_{14}	X_{23}	X_{33}	X_{44}	X_{34}	X_{23}	X_{14}	X_{06}
X_{06}	X_{15}	X_{24}	X_{34}	X_{45}	X_{35}	X_{24}	X_{15}	
			X_{35}					

Given the resource digraph of Figure 4.1, and the initial resource set U_0 of example 4.1, the sets $X_4 = \phi^4 U_0$, $X_3 = \beta\phi^4 U_0$, $X_2 = \beta^2\phi^4 U_0$, $X_1 = \beta^3\phi^4 U_0$, and $X_0 = \beta^4\phi^4 U_0$ solve the problem of identifying what needs to be produced at each stage in order to possibly generate the largest final stage resource set. The elements under each column heading denote the image of U_0 under the map at the head of the column, for example, $\beta^2\phi^4 U_0 = \beta(\beta(\phi(\phi(\phi(\phi(U_0))))) = \{x_{22}, x_{23}, x_{23}\}$. These sets are formed in the order $\phi(U_0), \phi(\phi(U_0)), \dots, \beta(\phi(\phi(\phi(\phi(U_0)))))$, and so forth.

4.2 THE MANUFACTURING DIGRAPH APPROACH

A multi-stage production process can be represented by a digraph having an alternative structure to that of the resource digraph. This digraph, termed a manufacturing digraph here can be converted to a corresponding resource digraph although the conversion process is not always reversible; that is, given a resource digraph it may not be possible to determine uniquely the manufacturing digraph with which it is associated. Figures 4.2 and 4.3 illustrate a manufacturing and corresponding resource digraph respectively.

The motivation for introducing the manufacturing digraph is that the problem of determining what needs to be produced at each stage of a multi-stage production process in order to generate the largest possible resource set from the final production stage can also be obtained from a manufacturing digraph. Furthermore, the manufacturing digraph obviates the need for construction of a possibly complex resource digraph. Solution to the problem introduced in section 4.1 is presented in this section in terms of a manufacturing digraph.

A manufacturing digraph differs from a resource digraph in that it includes the manufacturing processes at each production stage. The following interpretation is given to the manufacturing digraph. The operation of any process requires the availability of its precursors (that is, resources used as input to a process within a production stage), thus enabling the production of its

successors (that is, resources output by the process within a production stage). Further, an intermediate or final resource becomes available provided it is produced by at least one process capable of producing it. Using this interpretation the resource element x_{12} , in Figure 4.2, output by the first production stage, can be produced by either process y_{11} or y_{12} . All other elements output at the first production stage can be produced in only one way (for example, x_{11} is produced by y_{11} , x_{13} and x_{14} by y_{12} , and x_{15} by y_{13}).

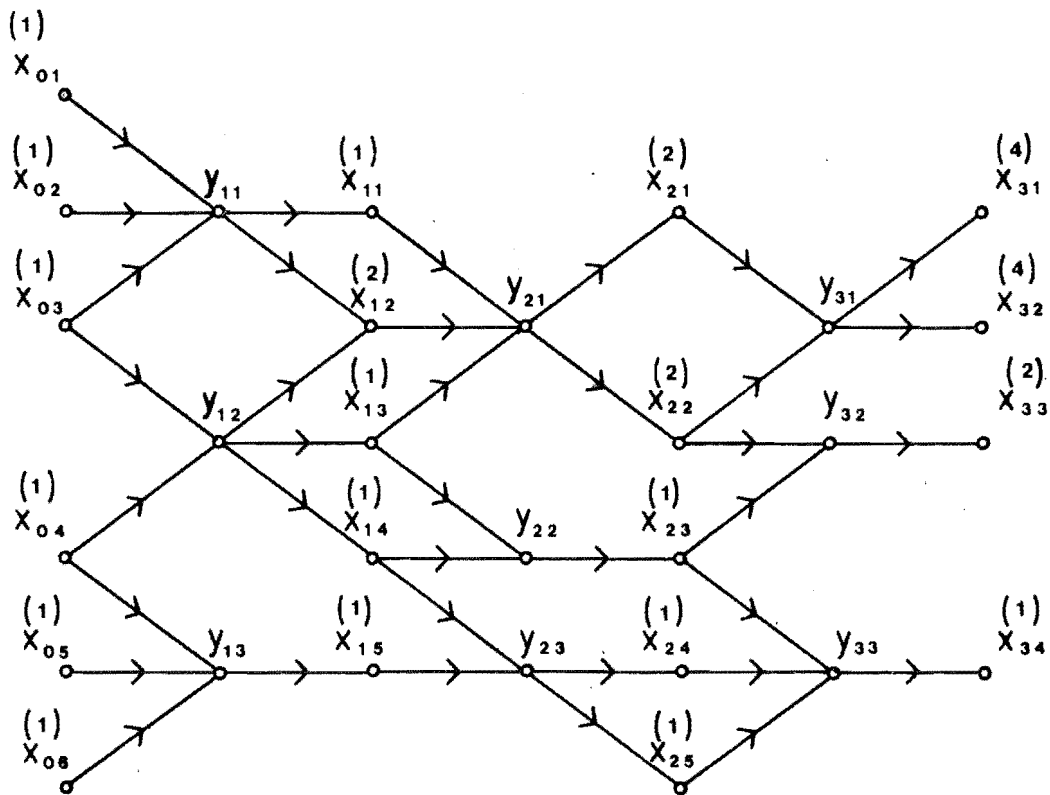


Figure 4.2 : Manufacturing Digraph Representing a Three-stage Process

where

x_{ij} = resource element j , either used as input to stage $i+1$, or output from stage i , or both.

y_{ik} = manufacturing process k , at production stage i .

The precursors of y_{ik} are $\{x_{i-1,j}\}$ and the successors $\{x_{i,j}\}$. The availability indices are indicated in parenthesis above the x_{ij} .

The construction of the resource digraph in Figure 4.3 from the manufacturing digraph in Figure 4.2 initially requires the determination of the numbers given in parenthesis above each resource vertex in Figure 4.2. These shall be termed "availability indices" in that they indicate the number of ways the resource element may be formed, given a complete set of initial resources and the multi-stage processes given in the manufacturing digraph. Initial resources are assigned availability indices of unity. For all other resources, the availability index is the sum, over all processes that produce the resource, of the product of the availability indices for precursors of that process. (For example, x_{12} may be produced by y_{11} or y_{12} , y_{11} has precursors x_{01} , x_{02} , x_{03} each having availability index 1. Their product is thus 1. The number of ways x_{12} can be produced through y_{11} is therefore 1. Similarly, y_{12} has precursors x_{03} and x_{04} , the product of the availability indices being 1. Thus the number of ways x_{12} can be produced by y_{12} is 1, and the total number of ways of producing x_{12} is $1+1 = 2$.

All resource vertices in the manufacturing digraph are labelled, the availability index indicates how many vertices in the appropriate stage of the resource digraph must have the corresponding label. (For example, examination of the availability indices of Figure 4.2 indicates that output from the second stage processes requires that the resource digraph include 2 vertices labelled x_{21} , 2 labelled x_{22} , 1 labelled x_{23} , 1 labelled x_{24} , and 1 labelled x_{25} . These vertices are contained in Figure 4.3. When an intermediate or final product can be produced in more than one way,

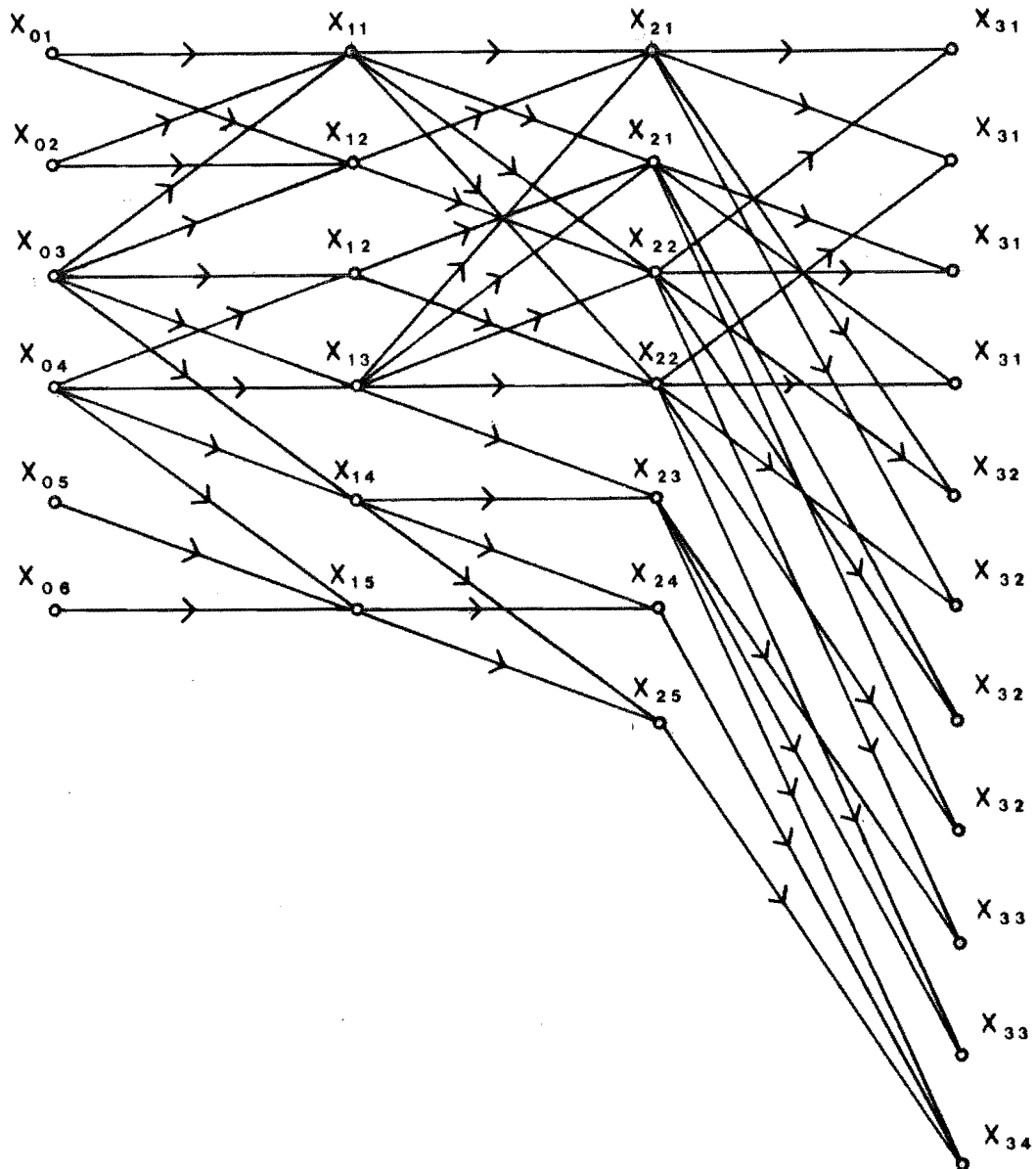


Figure 4.3 : Resource Digraph Representing a Three-stage Process

where

x_{ij} is resource element j , either used as input to stage $i+1$, or output from stage i , or both.

Vertex labels are repeated when alternative means of producing that resource exist, in this way the desired interpretation of the resource digraph is maintained, e.g. repetition of x_{12} means that x_{12} can be formed from either $\{x_{01}, x_{02}, x_{03}\}$ or $\{x_{03}, x_{04}\}$.

vertex labels must be repeated in the resource digraph in order that the correct definition be given when interpreting the resource digraph. This interpretation requires that all precursors of a resource element be available before formation of that element can take place. Repetition of a vertex label allows for formation of an element that can be formed from possibly different sets of precursors.

Once the availability indices have been determined the number of times each vertex label must be repeated in the construction of the resource digraph is available. The vertices used in construction of the resource digraph may be arrayed in stages, as in Figure 4.3, and arcs may be constructed between stages to represent each possible way of forming intermediate and final resources from the available resources and processes. Each possible way in which resource formation occurs is included in the resource digraph exactly once.

The following advantages accrue from representing a multi-stage process as a manufacturing digraph:

1. For each process, the resources required as input, and those generated as output, are indicated;
2. Intermediate or final resources produced in more than one way can be represented succinctly;
3. The relationship between processes and production stages is immediately apparent.

The most important of these is item (2) which allows for a terse description of the manufacturing process when intermediate or final resources are formed in more than one way. Constructing the resource digraph in this situation is made difficult because the number of arcs and vertices required quickly escalates in response to the number of

alternate ways of producing intermediate or final resources.

The manufacturing digraph $D(V,A)$ has the following structure. For an n stage production process, the vertex set V is partitioned into $2n+1$ subsets. These are the subsets V_0^* , P_1 , V_1^* , P_2, \dots, P_n, V_n^* where each of V_0^* , V_1^* , \dots, V_n^* is a set of distinctly labelled vertices (c.f. resource digraph). The set V_{i-1}^* for $2 \leq i \leq n$ usually denotes the set of resource elements produced by production stage $i-1$, from which input to production stage i is drawn. The vertex set P_i denotes the set of processes at stage i , precursors of which can be found in V_{i-1}^* and successors in V_i^* . Arcs of D either join a vertex in V_{i-1}^* to a vertex in P_i or join a vertex in P_i with a vertex in V_i^* . The set V_0^* consists of sources and the set V_n^* consists of sinks.

The problem of identifying what needs to be produced at each production stage in order to generate the largest possible resource set from the final production stage can be posed for the manufacturing digraph. This is analogous to the problem posed for the resource digraph. The problem is, given a set $U_0 \subseteq V_0$ of resource elements used as input to the first production stage, identify X_0^* , X_1^* , \dots, X_n^* subsets of V_1^* , V_2^* , \dots, V_n^* respectively such that the following hold.

1. $X_0^* \subseteq U_0$
2. If $x_i \in X_i^*$ for $1 \leq i \leq n$

then (a) $\exists y_k \in P_i \rightarrow y_k$ is a precursor to x_i
 (b) All precursors to y_k are in X_{i-1}^* .

3. If $x_i \in X_i^*$ for $0 \leq i \leq n-1$

then (a) $\exists y_k \in P_{i+1} \rightarrow y_k$ is a successor to x_i
 (b) All precursors of y_k are in X_i^* .

4. X_n^* is the largest possible.

A solution to this problem will enable a solution to be obtained to the related problem of determining whether a set $W \subseteq V_n^*$, output from the final production stage, can be produced from a set $U_0 \subseteq V_0^*$. This latter problem is of interest in that W could be a required product set, that is, a set that a facility must be capable of producing if it is to begin operation. The set U_0 could be the set of inputs available to the facility. Unless W can be produced from U_0 , operation of the facility will not be considered. One obvious solution to this problem that is contingent on a solution to the problem specified in 1. through 4. above is that, if $W \subseteq X_n^*$, then W may be produced from U_0 . Another solution method is given in section 4.3

One approach to determining the sets X_i^* in 1. through 4. above would be to express the manufacturing digraph as a resource digraph and use the forward and backward mapping procedures presented in section 4.1. The alternative method, developed in this section, is that, since any element in the image of the forward and backward maps ϕ and β will also be in the image of composite maps defined for the manufacturing digraph, then these latter maps may be used, thus obviating the need to construct the corresponding resource digraph. This latter approach requires the introduction of the maps ϕ_x , ϕ_y , β_x , and β_y and establishment of their properties via theorems 4.1 and 4.2 and corollaries 4.1 and 4.2.

The forward map $\phi_x : \mathcal{P}(V_i^*) \rightarrow \mathcal{P}(P_{i+1})$ maps subsets of V_i^* to subsets of P_{i+1} for $0 \leq i \leq n-1$. For $X_i \subseteq V_i^*$ the rule (4.4) defines ϕ_x .³

$$\phi_x(X_i) = \{v : v \in P_{i+1}, \text{ all precursors of } v \text{ are in } X_i\} \quad (4.4)$$

³ Because all vertices in a manufacturing digraph have a distinct set of labels, the images of ϕ_x, ϕ_y, β_x and β_y will always correspond to a vertex set with distinct labels.

The forward map $\phi_y : \mathcal{P}(P_i) \rightarrow \mathcal{P}(V_i^*)$ maps subsets of P_i to subsets of V_i^* for $1 \leq i \leq n$. For $Y_i \subseteq P_i$ the rule (4.5) defines ϕ_y . This rule implies there exists an arc (y_i, v) for some $y_i \in Y_i$. Thus, $\phi_y(Y_i)$ consists of the set of successors to Y_i .

$$\phi_y(Y_i) = \{v : v \in V_i^*, \text{ at least one precursor of } v \text{ is in } Y_i\} \quad (4.5)$$

The backward map $\beta_x : \mathcal{P}(V_i^*) \rightarrow \mathcal{P}(P_i)$ maps subsets of V_i^* to subsets of P_i for $1 \leq i \leq n$. For $X_i \subseteq V_i^*$ the rule (4.6) defines β_x .

$$\beta_x(X_i) = \{u : u \in Y_i \subseteq P_i, u \text{ is a precursor of any } x_i \in X_i\} \quad (4.6)$$

This map is restricted to points that lie in a subset Y_i in the codomain. The set Y_i is the set of processes under consideration and results from the mapping $\phi_x(X_{i-1})$.

The backward map $\beta_y : \mathcal{P}(P_i) \rightarrow \mathcal{P}(V_{i-1}^*)$ maps subsets of P_i to subsets of V_{i-1}^* for $1 \leq i \leq n$. For $Y_i \subseteq P_i$ the rule (4.7) defines β_y .

$$\beta_y(Y_i) = \{u : u \in X_{i-1}, u \text{ is a precursor of } y_i \in Y_i\} \quad (4.7)$$

Thus (4.7) consists of the set of precursors of Y_i .

Theorem 4.1 Let X_i^* be the distinctly labelled vertices formed from X_i (that is, $x_j \in X_i \subseteq V_i$ implies $x_j \in X_i^* \subseteq V_i^*$) then $x_j \in \phi(X_i)$ iff $x_j \in \phi_y(\phi_x(X_i^*))$ for $0 \leq i \leq n-1$.

Proof

The theorem states that the image set of $\phi(X_i)$ contains the same vertex labels as does the image set of $\phi_y(\phi_x(X_i^*))$. Note these sets are not necessarily equivalent since vertex

labels may be repeated in $\phi(X_i)$, in which case $\text{Card}(\phi(X_i)) \geq \text{Card}(\phi_y(\phi_x(X_i^*)))$.

Assume $x_{i+1} \in \phi(X_i)$, then by definition of ϕ the precursors of x_{i+1} exist in X_i . Construction of the manufacturing digraph ensures \exists at least one process $y_k \in P_{i+1}$ whose precursors can be found in X_i^* , i.e. $y_k \in \phi_x(X_i^*)$, and whose products include x_{i+1} , i.e. $x_{i+1} \in \phi_y(\phi_x(X_i^*))$.

Conversely, assume $x_{i+1} \in \phi_y(\phi_x(X_i^*))$. This then implies the existence of a least one process $y_k \in P_{i+1}$ in the manufacturing digraph producing x_{i+1} , whose precursors are found in X_i^* . By construction of the resource digraph \exists at least one way to produce x_{i+1} that is, $x_{i+1} \in \phi(X_i)$. \square

Theorem 4.2 Let X_i^* be the distinctly labelled vertices formed from X_i , then $x_j \in \beta(X_i)$ iff $x_j \in \beta_y(\beta_x(X_i^*))$ for $1 \leq i \leq n$.

Proof

The theorem states that the image set of $\beta(X_i)$ contains the same vertex labels as does the image set of $\beta_y(\beta_x(X_i^*))$. Note these sets are not necessarily equivalent since vertex labels may be repeated in $\beta(X_i)$, in which case $\text{Card}(\beta(X_i)) \geq \text{Card}(\beta_y(\beta_x(X_i^*)))$, here Card denotes cardinality.

Assume $x_{i-1} \in \beta(X_i)$, then x_{i-1} is a precursor to at least one vertex $x_i \in X_i \subseteq V_i$. This implies that x_{i-1} may be used, perhaps with additional precursors, in the manufacture of x_i . Construction of the manufacturing digraph ensures $\exists y_k \in P_i \ni y_k$ produces $x_i \in X_i^*$, $x_i \in X_i$ implies $x_i \in X_i^*$, i.e. $y_k \in \beta_x(X_i^*)$, and among whose precursors is x_{i-1} that is, $x_{i-1} \in \beta_y\beta_x(X_i^*)$.

Conversely, assume $x_{i-1} \in \beta_Y(\beta_X(X_i^*))$; then x_{i-1} is a precursor to at least one process y_k producing $x_i \in X_i^*$. Because $x_i \in X_i$, construction of the resource digraph ensures x_{i-1} is a precursor to $x_i \in X_i$, i.e. $x_{i-1} \in \beta(X_i)$. \square

Corollary 4.1

$x_j \in \phi^m(X_0)$ iff $x_j \in (\phi_Y \phi_X)^m(X_0^*)$ for $1 \leq m \leq n$, $X_0 \subseteq V_0$ and $X_0^* \subseteq V_0^*$, X_0^* consists of the distinctly labelled vertices of X_0 , x_j is a vertex label.

Proof (by induction)

$P(1)$ is true, since $x_j \in \phi(X_0)$ iff $x_j \in \phi_Y(\phi_X(X_0^*))$ holds by theorem 4.1 with $i=0$.

Assume $P(k)$ true for $1 \leq k \leq n-1$; then

$$x_j \in \underbrace{\phi(\phi(\dots\phi(X_0)\dots))}_{k(\phi\text{'s})} \text{ iff } x_j \in \underbrace{\phi_Y(\phi_X(\phi_Y(\phi_X(\dots\phi_Y(\phi_X(X_0^*))\dots)))}_{k(\phi_Y\phi_X\text{'s})}$$

Let $X_k = \phi(\phi(\dots\phi(X_0)\dots))$ then $X_k \subseteq V_k$ by definition of ϕ .

Let $X_k^* = \phi_Y(\phi_X(\phi_Y(\phi_X(\dots\phi_Y(\phi_X(X_0^*))\dots)))$ then $X_k^* \subseteq V_k^*$ by definition of ϕ_X and ϕ_Y .

Consider $\phi(X_k)$ and $\phi_Y(\phi_X(X_k^*))$, then $x_j \in \phi(X_k)$ iff $x_j \in \phi_Y(\phi_X(X_k^*))$ as this is theorem 4.1 with $i=k$. Thus $P(k+1)$ holds. \square

Corollary 4.2

$x_j \in \beta^{n-i}\phi^n X_0$ iff $x_j \in (\beta_Y \beta_X)^{n-i}(\phi_Y \phi_X)^n X_0^*$ for $0 \leq i \leq n-1$, where x_j denotes a vertex labelled x_j . $X_0 \subseteq V_0$, $X_0^* \subseteq V_0^*$ and X_0^* is the distinctly labelled vertices of X_0 .

Proof (by induction)

$x_j \in \phi^n X_0$ iff $x_j \in (\phi_Y \phi_X)^n X_0^*$, this being corollary 4.1 with $m=n$. Let $X_n = \phi^n U_0$ and $X_n^* = (\phi_Y \phi_X)^n X_0^*$, then proof of the corollary requires $x_j \in \beta^{n-i} X_n$ iff $x_j \in (\beta_Y \beta_X)^{n-i} X_n^*$.

$P(1)$ is true because X_n and X_n^* contain the same vertex labels and by theorem 4.2 $x_j \in \beta(X_n)$ iff $x_j \in \beta_y(\beta_x(X_n^*))$.

Assume $P(k)$ true for $1 \leq k \leq n-1$, then

$$\begin{aligned} & \overbrace{x_j \in \beta(\beta(\dots\beta(X_n)\dots))}^k \text{ iff} \\ & x_j \in \beta_y(\beta_x(\beta_y(\beta_x(\dots\beta_y(\beta_x(X_n^*))\dots)))) \end{aligned}$$

Let $X_{n-k} = \beta(\beta(\dots\beta(X_n)\dots))$ then $X_{n-k} \subseteq V_{n-k}$ by definition of β .

Let $X_{n-k}^* = \beta_y(\beta_x(\beta_y(\beta_x(\dots\beta_y(\beta_x(X_n^*))\dots))))$ then $X_{n-k}^* \subseteq V_{n-k}^*$ by definition of β_x and β_y . Consider $\beta(X_{n-k})$ and $\beta_y(\beta_x(X_{n-k}^*))$

then by theorem 4.2 with $i=n-k$, $x_j \in \beta(X_{n-k})$ iff

$$x_j \in \beta_y(\beta_x(X_{n-k}^*)). \quad \text{Thus } P(k+1) \text{ holds.} \quad \square$$

Using corollary 4.2, it is possible to identify what needs to be produced at each production stage in order to generate the largest possible resource set from the final production stage. Jointly, corollaries 4.1 and 4.2 suggest that the resource digraph need not be formed. Instead, the manufacturing digraph and the maps ϕ_x , ϕ_y , β_x and β_y suffice. The approach is illustrated in Example 4.2.

Example 4.2

Given the manufacturing digraph of Figure 4.2 and the initial resource set $U_0 = \{x_{03}, x_{04}, x_{05}, x_{06}\}$, determine what needs to be produced at each production stage in order to output the largest set from the final production stage.

The forward maps are used in the determination of the largest set output from the final production stage, i.e.

$$X_3^* = (\phi_y \phi_x)^3 U_0 = \{X_{34}\}, \text{ while the backward maps identify}$$

what needs to be produced at each intermediate production stage in order that X_3^* can be constructed. The sets

identified are:

forward maps

 U_0 $\phi_x U_0$ $\phi_y \phi_x U_0$ $\phi_x \phi_y \phi_x U_0$ $\phi_y \phi_x \phi_y \phi_x U_0$ $\phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$

 $X_{03} \ Y_{12} \ X_{12} \ Y_{22} \ X_{23} \ Y_{33} \ X_{34}$ $X_{04} \ Y_{13} \ X_{13} \ Y_{23} \ X_{24}$ $X_{05} \quad X_{14} \quad X_{25}$ $X_{06} \quad X_{15}$ backward maps

 $\beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\beta_y \beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\beta_x \beta_y \beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\beta_y \beta_x \beta_y \beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\beta_x \beta_y \beta_x \beta_y \beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$ $\beta_y \beta_x \beta_y \beta_x \beta_y \beta_x \phi_y \phi_x \phi_y \phi_x \phi_y \phi_x U_0$

 $Y_{33} \ X_{23} \ Y_{22} \ X_{13} \ Y_{12} \ X_{03}$ $X_{24} \ Y_{23} \ X_{14} \ Y_{13} \ X_{04}$ $X_{25} \quad X_{15} \quad X_{05}$ X_{06}

The column tabulated under the leftmost character of each map is the image of U_0 under the map.

$$\begin{aligned}
X_2^* &= (\beta_Y \beta_X) (\phi_Y \phi_X)^3 U_0 = \{x_{23}, x_{24}, x_{25}\}, \\
X_1^* &= (\beta_Y \beta_X)^2 (\phi_Y \phi_X)^3 U_0 = \{x_{13}, x_{14}, x_{15}\}, \text{ and} \\
X_0^* &= (\beta_Y \beta_X)^3 (\phi_Y \phi_X)^3 U_0 = \{x_{03}, x_{04}, x_{05}, x_{06}\} \quad \parallel
\end{aligned}$$

Example 4.2 illustrates the production of a resource at an intermediate production stage, which is not utilised by the subsequent production stage. This is resource element x_{12} , output from the first production stage (see Figure 4.2). The backward map excludes x_{12} because only resources that can be used as inputs to the second stage process set under consideration are chosen.

This illustrates that the solution obtained for the problem, the sequence of sets $X_0^*, X_1^*, \dots, X_n^*$ is such that X_i^* for $1 \leq i \leq n-1$ is not necessarily the largest possible set. That is, X_i^* may not satisfy $X_i^* = \phi_Y(\phi_X(X_{i-1}^*))$ but instead may only satisfy $X_i^* \subset \phi_Y(\phi_X(X_{i-1}^*))$. In practice, this means X_n^* may be a strict superset for output that can actually be produced by the final production stage. This follows because if production of intermediate resources that cannot be used by the subsequent production stage ceases, then, depending on the types of production processes used, it may not be possible to produce the full complement of intermediate resources required for production of the final stage output.

4.3 THE GENERATOR SET APPROACH

There are two simple methods to check for the production of a set $W \subseteq V_n$ at the final stage of an n stage production process, when an initial set of resources $U_0 \subseteq V_0$ has been made available.

1. Construct the image of U_0 under ϕ^n by the forward mapping procedure and ensure $W \subseteq \phi^n U_0$.

2. Identify possible subsets of U_0 whose joint image under ϕ^n contains W .

The first of these approaches involves constructing the image set of U_0 under ϕ^n and then searching the image set to ensure that the set W is contained within it. Either mapping procedure contained in section 4.1 or 4.2 may be used to construct $\phi^n U_0$. If the mapping ϕ^n is many to one, as is the case when intermediate products can be formed in more than one way, constructing the image of ϕ^n will involve repetition because certain subsets of U_0 will have the same image in $\mathcal{P}(V_n)$. It is this repetition that motivates the search for the second method of ensuring the production of W .

The mapping ϕ^n has domain $\mathcal{P}(V_0)$, subsets of the first stage inputs and codomain $\mathcal{P}(V_n)$, subsets of the last stage outputs. The problem of determining whether a set W can be generated given a first stage input set U_0 can be stated in terms of a possibly restricted mapping, as does the image of U_0 under ϕ^n contain W . If the mapping ϕ^n is one to one or many to one, then any set in the codomain may be formed as the image of possibly more than one set in the domain of ϕ^n . If the sets in the codomain are restricted to being individual elements, then the minimal sets in the domain that map on to these elements are the generator sets subsequently introduced.

If ϕ^n is a many-to-one map, then this suggests that W may be formed in a number of ways. This section identifies a special class of subset in the domain of ϕ^n such that W may be constructed from the image of these subsets under ϕ^n .

Definition 4.1⁴

A generator (set) for a final resource element $w_i \in V_n$ is defined to be an initial resource set $U = U(w_i) \subseteq V_0$ such that, the following are satisfied.

1. $\phi^n U_0 \supseteq w_i$
2. $\phi^n (U_0 \setminus \{u_j\}) \not\supseteq w_i \quad \forall u_j \in U_0$.

The first of these requirements ensures that $U(w_i)$ contains the initial resources required to produce a set containing w_i , while the second enquires that no smaller subset of $U(w_i)$ can be found that also produced w_i . Generally, any $w_i \in V_n$ may have more than one generator set. This can arise because different process sequences consuming different resources may be used to produce the same final resource. Denote the set of generators for an element $w_i \in V_n$ by the set (4.8)

$$U(w_i) = \{U_k(w_i)\} \text{ for } k = 1, \dots, n_{w_i} \quad (4.8)$$

where

n_{w_i} is the number of generators for resource element $w_i \in V_n$.

There must always exist at least one generator for each w_i . Otherwise, w_i is not part of a production possibility set. Generator sets are best identified from the resource digraph by starting with $w_i \in V_n$ and tracing back over all backward paths emanating from $w_i \in V_n$ and terminating in vertices contained in V_0 . The set of vertices so described will give a generator set, $U(w_i)$ for w_i . Different generator sets arise when distinct vertices having the same label in V_n trace back to different vertex sets in V_0 . These generator sets may be used to check for the production of a

⁴ It is possible to directly define generator sets for a subset $W \subseteq V_n$ in an analogous manner to definition 4.1. This was the approach adopted in the test problem of section 5.1.

set $W \subseteq V_n$ at the final production stage; the following theorem is required.

Theorem 4.3 If $W \subseteq V_n$ and $\bigcup_{w_i} U(w_i) \subseteq U_0 \subseteq V_0$ then

$$W \subseteq \bigcup_{w_i} \phi^n(U(w_i)) \subseteq \phi^n(\bigcup_{w_i} U(w_i)) \subseteq \phi^n U_0$$

where set unions are taken over all elements $w_i \in W$.

Proof (omitted)

The proof follows quite simply from the properties of ϕ^n , $U(w_i)$, and the transitivity of the relation \subseteq . \square

Theorem 4.3 states that the initial resource set $U(U(w_i))$, $w_i \in W$ is sufficient to produce W (it may not be necessary to have all elements from this set in order to produce W - a strict subset may suffice). This set is just the union of the generator sets for $w_i \in W$. However, this union may be made up in more than one possible way when more than one generator set exists for any $w_i \in W$. The product (4.9) denotes the number of possible ways of forming this union

$$\prod_{w_i} n_{w_i} \text{ for } w_i \in W \quad (4.9)$$

Provided the product specified in (4.9) is small in relation to the number of elements in W , then theorem 4.3 is easy to apply, and a sufficient condition that $W \subseteq V_n$ can be produced from $U_0 \subseteq V_0$ is that $\bigcup_{w_i} U(U(w_i)) \subseteq U_0$. If this condition is not met W may still be produced from U_0 . This is a limitation to application of generator sets defined in this way. Example 4.3 illustrates the application of generator sets.

Example 4.3

Given the resource digraph of Figure 4.3, identify generator sets $U(w_i)$ for $w_i \in V_3$, then using theorem 4.3

determine whether $U_0 = \{x_{01}, x_{02}, x_{03}, x_{04}, x_{05}\}$ is sufficient to produce $W = \{x_{31}, x_{32}, x_{33}, x_{34}\}$

$w_i \in V_n$	generator sets $U(w_i)$
x_{31}	$U_1(x_{31}) = x_{01}, x_{02}, x_{03}, x_{04}$
x_{32}	$U_1(x_{32}) = x_{01}, x_{02}, x_{03}, x_{04}$
x_{33}	$U_1(x_{33}) = x_{01}, x_{02}, x_{03}, x_{04}$
x_{34}	$U_1(x_{34}) = x_{03}, x_{04}, x_{05}, x_{06}$

In this case, there is only a single generator for each $w_i \in V_n$, forming the possible unions of the generator sets is simple as there is only one such union.

$$\bigcup_{w_i} U(w_i) = \{x_{01}, x_{02}, x_{03}, x_{04}, x_{05}, x_{06}\}$$

Because this is not a subset of U_0 , the conditions of theorem 4.3 are not met and thus there is no guarantee of the production of W from U_0 . Application of the forward mapping procedure shows that W cannot be produced from U_0 . //

4.4 IMPLEMENTATION OF SET MAPPING PROCEDURES

An implementation of mapping procedures described in this chapter to detect redundancy in a multi-stage production process is envisaged as follows.

Consider a planning problem wherein the future operations over a planning period consisting of a finite set of model periods of a processing facility that utilises forest roundwood is being examined. Provided that an adjoining set of model periods, inclusive of the last model period can be found, during which the mapping procedure

(either the procedure from section 4.1, involving the resource digraph, or that from section 4.2 involving a manufacturing digraph, could be used) suggests that the facility may be capable of producing its specified production set, then the facility may be incorporated as part of the planning problem during this set of model periods. Otherwise, the facility can be dismissed from further consideration, in that the theory of this chapter indicates that it will never meet the required criterion for operation, that of producing a specified production set.

The construction of arcs, representing the transportation of resources input to, and output from, a facility can be made contingent upon the inclusion of the facility as part of the planning problem. When this is done, it means arcs are not added to the problem unless there is a possibility of their utilisation.

CHAPTER 5

COMPUTATIONAL EXPERIENCE

Solution to Forest Utilisation Management Problems (FUMPs) requires the pattern of resource flows between processing centres to be established over time. These resource flows incorporate the transfer of roundwood from forests and may subsequently include the transfer of processed roundwood. Regulatory aspects of initiating roundwood flows were discussed in chapter 2, while chapter 3 and 4 dealt with problems attendant with roundwood and processed roundwood flows, and the resolution of these problems using mathematical programming techniques.

This chapter contains details of a test problem that uses regulatory mechanisms, previously introduced in this study, to control resource flows between processing centres. Specifically, the forest regulatory mechanisms of section 2.3 are employed to regulate roundwood production. A non-convex introduction mechanism from section 3.3 is used to model the possible operation of sawmills of differing capacities.

Computational aspects of this problem are discussed, notably the requirements of matrix generators for problem specification and the tabulation of results using report writers. Additionally, material detailing difficulties associated with mathematical programming techniques in forest planning is also incorporated. Note that these difficulties are faced in lesser or greater degree in forest planning models employing techniques other than that of optimisation.

An outline of the sections is as follows. Section 5.1 presents a brief description of the test problem, along with related requirements of the data base and solution documentation phase, also problems encountered during the solution of the test problem are included. Section 5.2 discusses development and validation procedures for matrix generators, along with making largely historical observations on generators and report writers. Section 5.3 discusses difficulties encountered in the application of mathematical programming techniques to forest planning.

5.1 DESCRIPTION OF TEST PROBLEM

The general nature of the test problem is initially described, then data input requirements are identified in subsection 5.1.1, report writing tabulation procedures are dealt with in subsection 5.1.2, and reporting on singularities that were indicated during the solution of the test problem is covered in subsection 5.1.3.

An example of the Forest Utilisation Management Problem is to consider the activities of a hypothetical forest products company involved in regulating roundwood supplies and considering the possible expansion to utilisation operations by constructing one or more sawmills. Note that the test problem considered in this study is only one of many possible problems from the class of FUMPs.

The forest products company considered is currently entirely dependent upon roundwood (log) sales for its revenue source. The company is interested in possibly expanding operations by constructing one or more sawmills to consume roundwood from forests regulated by the company and produce

various grades of sawn lumber, together with residues, which could be sold at any of two available markets.

To examine the feasibility of their proposals, the company adopts a planning period of 30 years partitioned into 5-year model periods. This, the company considers, will allow a basis for comparison of their existing operations with their proposed operations. Any decision to change their existing form of operation will be based largely on comparing the efficiencies of the existing operation with those of the expanded system. Should the proposed system appear more "efficient", an assessment of the risk involved in the proposal could then be used as a selection criterion by management.

The company's forests consist entirely of stands of *Pinus radiata* (D. Don) (note - the extension to the multiple species situation provides no conceptual difficulty). Existing stands comprise some 2050 hectares, the oldest stand being 35 years old. These stands comprise three crop-types (the yields from which conform to one yield table; Shirley, 1984) denoted 1, 2 and 3. Crop-type 1 denotes intensively tended stands aimed at the production of high-grade sawn timber; Crop-type 2 denotes medium tended stands aimed at the production of sawlogs yielding predominantly lower to middle grades of sawn lumber; Crop-type 3 denotes stands with low levels of tending, which yield low grades of sawn lumber. The company has roughly equal areas of each 5-year crop-type/age class, arising from previously conducted afforestation schemes.

Because of the company's future position as one of:

1. a log supplier;
2. a utiliser of its own logs;
3. both (1) and (2);

and considering the ready availability, from external sources, of low-grade logs. It considers further production of crop-type 3 not to be in its interests. All reafforestation of current stands is to be undertaken with either crop-type 1 or crop-type 2. The company has an option on an additional 380 hectares of currently unutilised land which, if purchased, will be afforested with either crop-type 1 or crop-type 2.

The company is uncertain as to whether its interests are best served by reafforesting (current management units) or afforesting (future management units) with crop-type 1 or crop-type 2. These decisions are contingent upon its future position, 1, 2 or 3 above. The company considers that the continued production of roundwood requires the productive capacity of their forest to be maintained. To ensure this, a residual volume of at least $100,000 \text{ m}^3 \text{ annum}^{-1}$ in harvestable age classes must be maintained.

The market structure for the company's roundwood sales is particularly simple, the company being unable to influence market price, although it is always guaranteed sale of its roundwood products. The company considers that a similar situation would apply to its position in a sawn lumber and residues market.

The forms of utilisation being proposed by the company are sawmills of differing capacities, the smaller having an annual roundwood input of $30,000 \text{ m}^3$, and the larger with an annual roundwood input of $60,000 \text{ m}^3$. Introduction costs show constant returns to scale with respect to capacity. They are \$2,000,000 and \$4,000,000 for the smaller and

larger mill respectively¹. The recovery factors for sawn lumber out-turn and residue production are the same for both facilities. However, unit processing costs (namely, \$m⁻³ roundwood crop log input) are slightly lower in the larger mill, in that there is some underutilisation of productive capacity in parts of the smaller mill where the operating costs incurred are largely independent of capacity.

Preliminary estimates indicate that an introduction of a larger mill may mean facing a shortfall in the roundwood required for the mill to operate at its annual capacity of 60,000 m³ because the forests regulated by the company are capable of a sustained annual production in the region of 50,000 m³.

Three possible sites (locations) exist where construction of the mills can take place. At any site, either facility (plant) may be constructed. The company considers capacity introductions may take place at any of six model periods. If introduced, these mills will receive roundwood from company-regulated forests, and products will be shipped to either the two lumber and residue markets that are available. These markets differ both in terms of their proximity to mill sites, and in their market structure, in that individual product prices differ at these markets.

The company considers that an efficiency measure can be constructed from costs and revenues incurred during the planning horizon in the following manner. The net present value of revenue gained by way of roundwood, lumber, and residue sales, during the planning horizon less the net present value of costs expended in the formation of these

¹

These values are artificially low so as to ensure that capacity introductions would occur and the functioning of the model components could be checked.

products and in guaranteeing future supplies constitutes a suitable economic efficiency measure. This measure is inclusive of payments made to introduce mills. It is also inclusive of any activity that involves a net revenue or net cost to the modelled system during the planning horizon.

(Note: transactions internal to the modelled system are neither net costs nor net revenues and always sum to zero). This measure does not require valuation of company stock at any stage during the planning horizon and is best considered as a measure of the earning power of the company's stock before taxation.

The general flow digraph for the test problem (c.f. chapter 3) is illustrated in Figure 5.1 as indicated in chapter 3, the solution to the FUMP will determine which flows represented in the digraph are to occur. The flows selected in the solution to the problem occur along the dashed arcs.

The Burroughs application program ^(BURROUGHS CORPORATION) (1976) was used to generate a representation of the test problem (that is, the MPS form) used as input to the Burroughs mathematical programming system TEMPO (1975). The MODELER program is file MG, and the data base file DB, in Appendix MYFILES. The data base is accessed by the matrix generator program and also by the report writer program (file RW in Appendix MYFILES) subsequent to problem solution.

The generated test problem comprised some 2523 columns, 36 of which were binary integer variables, and 1623 rows. Subsequent to generation, it was solved using the Branch and Bound algorithm available on the Burroughs TEMPO system. The machine used was the Burroughs B6920 machine at the University of Canterbury. This implementation of the branch and bound algorithm requires that the LP formed by

Fig. 5.1 (addendum) Major Constraint types utilised within the FUMP test problem.

Forest Estate Constraints:

Both current and future management units have an area partition of the following form. The equality form of the constraint corresponds to a supply driven situation.

$$\sum_{k=1}^{K_{ij}} \sum_{\ell=1}^{L_{ijk}} x_{ijkl} \quad \left\{ \begin{array}{l} \leq \\ = \end{array} \right\} A_{ij}$$

where

- x_{ijkl} are the units of area for the management unit defined by crop-type i , age class j , managed under strategy ℓ that uses crop-type k for reafforestation;
- A_{ij} are the units of area for the management unit defined in the initial period by crop-type i , age class j .

The following regulatory constraints were imposed on residual and harvest volume. The residual volume in harvestable age classes at each planning period was both smoothed and bounded using the following constraints.

$$(1 + \gamma_j) r_j - r_{j+1} \leq 0 \quad \forall j$$

$$r_0 \geq \psi^{\min}$$

where

- r_j is the residual volume in harvestable age classes at period j ;
- ψ^{\min} is a lower bound for the residual volume in harvestable age classes during the initial model period; and
- γ_j is a smoothing parameter such that $(1 + \gamma_j) > 0$.

Additionally harvest volume smoothing was undertaken using the following constraints.

$$(1 + \gamma'_j)r'_j - r'_{j+1} \leq 0 \quad \forall j \in J$$

where

r'_j is the harvest volume in period j ; and
 γ'_j is a smoothing parameter such that $(1 + \gamma'_j) > 0$.

Processing Facility Constraints:

The following introductory regulatory mechanism was employed.

$$S = \{y_{pt} : y_{pt} = 0 \text{ or } 1, \quad t \in T, \quad p \in P\}$$

$$y_{pt} = \begin{cases} 1 & \text{introduction of facility } p \text{ at the} \\ & \text{start of model period } t \\ 0 & \text{otherwise} \end{cases}$$

Resource production constraints:

These may be used to bound factors of production produced during operation of the facility. Analogous constraints for factors consumed may be used to specify infeed capacity restrictions.

$$\sum_{s < t} r_{pis} y_{ps} - \sum_{k \in K_{ip}} a_{pki} x_{ptk} \leq 0 \quad \forall p, i, t$$

$$\sum_{s < t} r_{pis}^* y_{ps} - \sum_{k \in K_{ip}} a_{pki} x_{ptk} \geq 0 \quad \forall p, i, t$$

where

- x_{ptk} is the activity level of process k in period t at facility p , $x_{ptk} \geq 0$;
- a_{pki} is the technological parameter indicating production of resource i per unit activity of process k at facility p ;
- $r_{pis} (r_{pis}^*)$ are the lower (upper) limits for the production of resource i in period s by facility p ; and
- K_{ip} is the set of processes at facility p that produce resource i .

The following constraints may be imposed as an alternative, or in addition to, the previous constraints. These constraints restrict output relative to total production (input restrictions can be developed analogously).

$$\sum_{k \in K_{pr}} a_{pkr} x_{ptk} - \alpha_r u_{pt} \leq 0 \quad \forall p, t, r$$

where

- u_{pt} is the total output of facility p in period t ;
- α_r is the percentage of total output subject to restriction r ;
- a_{pkr} is the technological parameter for output restriction r in process k at facility p ; and
- K_{pr} is the set of processes subject to restriction r at facility p .

Budget Constraint:

$$\sum_{p \in P} \sum_{s \in S} C_{pst} y_{ps} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} C_t \quad \forall t$$

where

C_{pts} is the payment incurred in period t to introduce facility p at the start of period s ; and

C_t is the bound on payments that may be made during model period t . This is associated with introductions during or prior to period t .

relaxation of integer restrictions on the MILP be initially solved by the TEMPO routine PRIMAL, before initiation of the branch and bound procedure, MXINT.

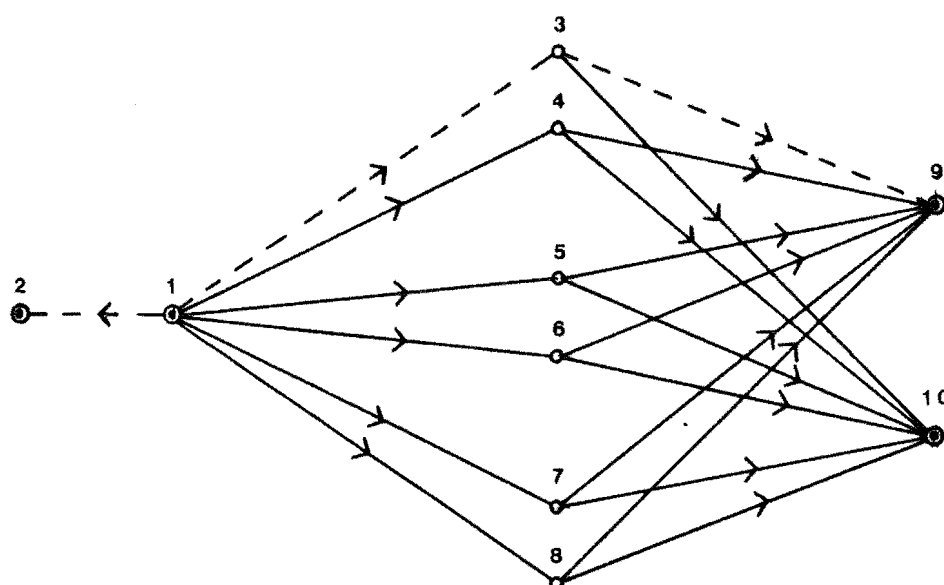


Figure 5.1 : General flow digraph illustrating option on continued log sales, and expansion of processing and marketing options. The vertices are labelled as follows²:

- | | |
|---------------------------------------|---------------------------------------|
| (1) Forest estate | (2) Roundwood market |
| (3) Site 0 30,000 m ³ mill | (4) Site 0 60,000 m ³ mill |
| (5) Site 1 30,000 m ³ mill | (6) Site 1 60,000 m ³ mill |
| (7) Site 2 30,000 m ³ mill | (8) Site 2 60,000 m ³ mill |
| (9) Lumber and residue market | (10) Lumber and residue market |

Dashed arcs represent flows between processing centres that were selected in the problem solution.

In solving the relaxed problem, PRIMAL took 10.56 minutes of Central Processor Unit (CPU) time to reach feasibility during Phase I of the simplex procedure. An additional 22.37 minutes CPU time were required to reach optimality for the relaxed problem. The branch and bound procedure was then initiated and took 37.39 minutes of CPU

² The following notation is used on the data base and report writer output for vertices labelled 1 through 10.

- | | |
|-----------------------------|-----------------------------|
| (1) Alpha 0 | (2) Tau 0 |
| (3) Site sigma 0 facility 1 | (4) Site sigma 0 facility 0 |
| (5) Site sigma 1 facility 1 | (6) Site sigma 1 facility 0 |
| (7) Site sigma 2 facility 1 | (8) Site sigma 2 facility 0 |
| (9) Omega 0 | (10) Omega 1 |

time to terminate, during which time, two feasible integer solutions were identified. To facilitate execution of the branch and bound procedure, an initial cut-off for the objective function was used (this was gained by prior solution of related problems). For the maximisation problem, this enabled candidate problems (and their descendants), whose relaxed solutions were lower than the cut-off value to be excluded from further consideration (see Burroughs TEMPO Manual, *op. cit.*, pp8-1 to 8-34). A complication that arose during the solution of this problem is discussed in subsection 5.1.3.

The solution to the test problem suggests that the company should utilise logs from the forest it regulates in a sawmill, as well as pursuing its already existent log sales. The suggested benefits for adopting this option is that the company would receive, over the planning horizon, revenue whose net present value is some \$2.6 million in addition to that it would gain by pursuing log sales alone. Suggested in the solution is that, the smaller-capacity sawmill should be introduced at site 0 (the vertex labelled 3 in figure 5.1) at the beginning of the second model period. The lumber and residue products being sold at only one market (the vertex labelled 9 in Figure 5.1).

The optimal solution indicates that lower-grade roundwood material from each of the crop-types should be sold on the roundwood market. Indeed, all roundwood from crop-type 3 should be disposed of in this manner, and only the higher quality roundwood logs from crop-types 1 and 2 should be used as input to the mill. Despite the higher operating costs of the smaller mill, its introduction is logical because the larger mill would be unable to operate

at full capacity given the current residual volume constraint and the productive capacity of the company's forest. Details from solution of the test problem are tabulated in file RP in Appendix MYFILES, this is the report produced by the report writing program, (file RW in Appendix MYFILES). No attempt was made to examine the sensitivity of the test problem solution by variation of model parameters.

5.1.1 Data Base Requirements

This subsection identifies the data base requirements for the Forest Utilisation Management Problem specified by the test problem. Considered are the data requirements in the areas of forest regulation and production along with utilisation transportation and marketing. The data requirements in each of these areas are largely distinct reflecting the two phases of a FUMP, that is, regulation and growth of roundwood, and its subsequent utilisation. Discussed are the different data requirements in each of these areas and the connection between the data base and the form of the model is implicitly indicated.

FOREST REGULATION AND YIELD DATA

Management unit definition allows the subsequent definition of a number of data items that relate to stand management within a unit. In considering management unit data, note that some quantities are defined for current or future management units only. The need for this should become apparent from the differing descriptions of current and future management units.

Current management units are those stocked prior to the planning horizon and are regulated as part of the forest

estate during the planning horizon. Future units are areas afforested during the planning horizon. The area regulation mechanism employed in the test problem was essentially a model I representation (see subsection 2.2.1), extended to deal with future management units. A model I management unit was described in subsection 2.2.1 as being an identifiable forest area associated with a crop-type/age class combination in the initial planning period. These are the current units, whereas the identifiable forest areas that become available for afforestation during the planning horizon are future management units.

Possible crop-types within current management units are considered in the following manner. Management alternatives allow for transitions to different crop-types only upon clearfelling of the crop-type that exists at the start of the first model period. On the other hand, each management alternative for future management units involves only a single crop-type. Different crop-types, however, may be considered in different alternatives. Thus, a future management unit possesses a set of management alternatives for each crop-type considered.

The structural constraints required for area regulation in current units are similar to those given in subsection 2.2.4, system (2.10). Extending this system to deal with future management units is straightforward. Alternatives for future management units consider management activities to begin in the model period in which afforestation takes place.

Data relating to forest regulation and recovery of roundwood will subsequently be discussed in this subsection. When dependencies exist between the data items and the form

of the management unit, these are indicated. An abridged index of the data base (file DB in Appendix MYFILES) is given in Appendix 5.1.1(A). It is abridged because it indexes data items required in the model but is exclusive of information required for the execution of the Matrix Generator program that is also tabulated in the data base. A guide to interpreting the tabulations in file DB is given in Appendix 5.1.1(B). All tables and lists subsequently referenced are contained in file DB in Appendix MYFILES.

Management Unit Area

The tables designated EXISTINGLANDAREA tabulate areas associated with current management units. These data define the initial crop-type/age class area partition introduced in subsection 2.2.1. Similarly, the table designated FUTURELANDAREA tabulates area associated with future management units.

Management Unit Terrain

The terrain of a management unit is classified according to the likely mode of harvesting to be employed. The classification used allows for the description of forest land by harvesting mode, volume may be recovered from a management unit by skidder and/or hauler harvesting operations. Data relating to this classification are tabulated in tables designated SR (SKID RATIO), a skid ratio indicates the proportionate area of a management unit permitting skidder harvesting operations.

Management Unit Thinning

The mechanism used to incorporate production thinning is best understood as a device that permits a partial harvest before clearfelling at the end of the crop rotation. Tables designated PRIORTHIN contain data that indicate whether production thinning has been undertaken on current management units before the start of the first planning period. The occurrence of thinning affects subsequent stand yields (see Figure 5.2). Data contained in tables designated UNIT_THIN are used to indicate which units are to be considered for production thinning. Thinning is restricted to stands on skidder terrain. Other factors used to determine whether a production thinning takes place are the crop-type and age class of the stand.

Management Unit Access

The requirements of access to management units to permit tending operations, and removal of wood resources, requires that roading networks be developed and maintained. Costs associated with roading activities are included within the database. Table ACCESSCOST contains, for units afforested during the planning horizon, costs of constructing initial access roading prior to afforestation. Table ROADCOST details the cost of upgrading access roading to permit the removal of harvest volumes. These costs are considered to be incurred one year before clearfelling in both current and future management units.

Management Unit Roundwood Recovery

Roundwood extraction costs incurred at harvest time for various crop-type, age class combinations are tabulated

in the data base for both skidder and hauler terrain types. These are the tables SKIDCOST and HAULCOST.

Management Unit Yield Tabulation

Management alternatives may have associated combinations of several partial fellings followed by clearfelling at various ages. Yield data must be tabulated so that recoverable volumes at any age can be specified. Measures of harvest or residual volume can be regarded as a resource set suitable for constraint. Generally, the inclusion of yield data into an FMP or FUMP involves forming discrete approximations to continuous growth curves, the approximation being a point estimate in each model period. Indicated in Figure 5.2 is a pair of (piecewise) continuous growth curves for successive rotations. A small time lapse occurs between rotations, during which, activities such as site preparation can occur.

One convenient way to incorporate partial fellings is to regard them as stand transitions where the crop-type site index remain the same. Only the stocking numbers are reduced. The first rotation in Figure 5.2 undergoes such a transition. Following the arrowed curve gives the yield for the stand (except at t^*). Crop-type yields may then be tabulated for stocking densities before and after production thinning. Production thinning volumes can be determined from the difference in yields at the time of production thinning.

This method forms the basis for tabulation of yield data in the data base. Separate tables are, however, provided for the initial crop-types and for those used in restocking current management units because the composition of crop-types may vary in subsequent rotations (in particular, a crop-type used in the model may be an aggregation of basic crop-types).

In the data base, these are designated EYDU, EYDL (that is, EXISTING YIELD UPPER STOCKING, EXISTING YIELD LOWER STOCKING) and RYDU, RYDL (that is, RESTOCKED YIELD UPPER STOCKING, RESTOCKED YIELD LOWER STOCKING). Yield tables for crop-types associated with future management units are designated FYDU and FYDL (that is, FUTURE YIELD UPPER STOCKING, FUTURE YIELD LOWER STOCKING).

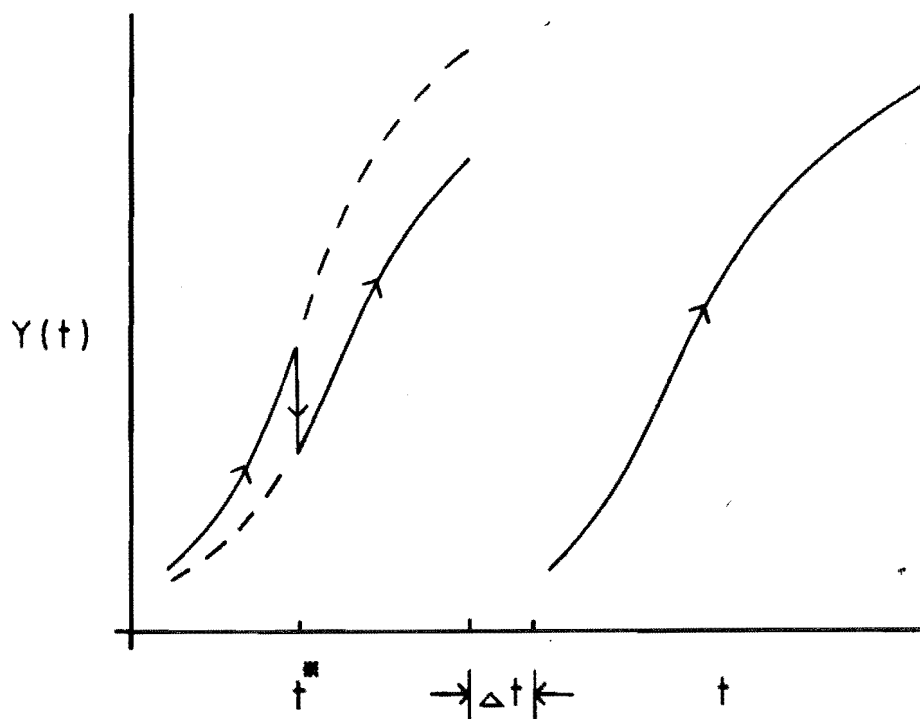


Figure 5.2 : Generalized Growth Curves in Even-aged Stands

$Y(t)$ is the yield at time t , the first rotation contains a production thinning at time t^* . There is a small delay between termination of the first rotation and the start of the second; this is the interval Δt .

Management Unit-Tending Costs

Associated with the growth of even-aged stands in plantation forests is a form of costs commonly called tending costs. They are, as their name implies, costs incurred in growing the crop up to the time it is harvested. Typically, tending costs include expenditure on

1. Land preparation;
2. Establishment;
3. Releasing and blanking;
4. Pruning and thinning operations;
5. Fertiliser application; and
6. Administration and protection.

Tabulation of tending costs at four discount rates (5% was used for the test problem) is given for various crop-type/age class combinations in the tables designated TENDCOST in the data base. The tabulation of discounted cost flow streams precludes the need to tabulate individual tending costs.

Roundwood Bucking Mechanism

The bucking procedure is implemented in conjunction with a grading system and provides the means to convert recoverable volume to roundwood volumes by log class. These log classes form the roundwood flows output from the company's forest. Each of the harvested crop-types is bucked (partitioned) into as many as five crop log classes. The mechanism used takes account of the age of the crop-type harvested, the stem position, and the length of logs. Lack of adequate data prevented consideration being given to other attributes that may be used to grade logs, notably diameter and log quality measures (Whiteside and Manley, 1985). Conceptually, the inclusion of these within a model structure provides little difficulty in terms of problem formulation, but it does increase markedly the associated problem of data collection for the processing phase. Processing activities at processing facilities are detailed with respect to log class inputs, thus the more log classes used, the greater the problem of specifying the associated processing activities. Data

relating to the bucking mechanism for the test problem is tabulated in the table designated BUCKPRAD (i.e. Buck *Pinus radiata*).

Management Alternative Generation

For stands that are homogeneous in the sense that they can be represented as a single crop-type, a management alternative for a model I formulation must consider the age of the stands and the timing of harvests. If the timing of harvests only is considered, then each management alternative gives rise to a *cutting pattern* that indicates when, if at all, harvests occur for that alternative. Figure 5.3 shows that the construction of management alternatives for different management units may involve repetition of certain sets (blocks) of cutting patterns. Application of this observation formed the basis of a generation procedure for the management alternatives in the extended model I formulation in the test problem.

If a set of crop-types has the same minimum and maximum ages of clearfelling, and if all crop-types used for afforestation or reafforestation are drawn from this set, then the coding of cutting patterns associated with management units is particularly simple. Firstly, blocks of cutting patterns where all members of a block have a common initial harvest period are defined. Next the cutting pattern with no harvests is added to these blocks. Each of these blocks is an element of a partition, and any temporal cutting pattern associated with a management alternative will be contained in one of these blocks. Figure 5.3 indicates that the temporal cutting patterns associated with a management unit can be regarded as a collection of such blocks. Given the temporal

management alternative	model period						block
	0	1	2	3	4	5	
E1	X				X		B0
E2	X					X	
E3	X						
E4		X				X	B1
E5		X					
E6			X				B2
E7				X			B3
E8					X		B4
E9						X	B5
D1	X				X		B0
D2	X					X	
D3	X						
D4		X				X	B1
D5		X					
D6			X				B2
D7				X			B3
D8					X		B4
D9						X	B5
D10							B>5

Figure 5.3 : Cutting Patterns for Two Type 1
Management Units

The repetition of the blocks B0 through B5 shows the temporal repetition of cutting patterns. Management alternatives E1-E9 and D1-D10 involve the stocking, and possible restocking of crop-types whose minimum and maximum ages of clear felling are 20 and 50 respectively. The ages of crop-types in strategies E1-E9 and D1-D10 in the first planning period are 30 and 25 respectively. The strategies for each unit are partitioned into a number of blocks by grouping alternatives that have the same first harvest period. Blocks B0 through B5 are common to both management units.

patterns, other attributes of the management alternative such as age, crop-type, terrain type, and so on can easily be incorporated so that the management alternative can be constructed fully.

The tables in the data base designated BLOCK-WALK identify for each species (only *Pinus radiata* crop-types were used in the test problem, however), blocks of cutting patterns. The tables designated INCLUDE-BLOCK specify, for each species, which blocks of cutting patterns are to be used to construct the management alternatives for a management unit.

Residual and Harvest Volume Regulation

Regulatory constraints, both bounding and smoothing (see section 2.3), were imposed in the test problem so as to examine the constraint formulations developed in section 2.3. Specifically, the residual volume in harvestable age classes at each planning period was both smoothed and bounded using the constraint system (2.19) and the inequality (2.29). In addition, harvest volume occurring during a connected subset of model periods (the last four periods of the six-period planning horizon) was smoothed using the system (2.19). In terms of the discussion in section 2.3, the bounding and smoothing of residual volume constitutes the imposing of a lower bound envelope for residual volume, while the smoothing of harvest volume constitutes the imposing of a partial envelope of smoothing constraints.

Data relating to the bounding and smoothing of residual volume is contained in the tables designated STAND_BOUND and STAND_SMOOTH. The table STAND_BOUND specifies the lower-bound on residual volume required by the

inequality (2.29) for the initial model period. Table STAND_SMOOTH specifies the scalars required for the system (2.19). Harvest volume smoothing data is contained in the tables STUMP_SMOOTH_TIMES and STUMP_SMOOTH. Data in first of these specifies which model periods are to have harvest smoothing constraints applied, while data in the second specifies the scalars required in the smoothing system.

TRANSPORTATION, UTILISATION AND MARKETING DATA

As indicated in section 3.2, transportation of resources is the means of linking joint productive processes at different geographic locations. Note that the general flow digraph presented in Figure 3.1 implies the existence of a network connecting processing centres. The utilisation of paths within this network is, as outlined in section 3.0, dependent on how possible resource transfers are governed.

The construction of arcs between processing centres and the tabulation of the required data is facilitated by the observation that generally resource flows between processing centres are acyclic. That is, it is not usually possible to create a loop amongst processing centres with the same resource, because a resource is not usually shipped to and then from a processing centre without undergoing further processing³.

The form of the data used in the construction of arcs is as follows. For each processing centre (geographically distinct), both the sets of centres from which resources may be input, and to which resources may be output, are

³ This suggests the cycle in the digraph in Figure 3.1 would refer to transfers of different resources at possibly different time periods.

identified. Furthermore, both the period in which the shipment possibly occurs and the identity of the material being transferred must be known. Jointly, this constitutes the data required to construct arcs for resources shipped between processing centres.

The required data are tabulated in the data base in tables designated FORESTTO; SITEFROM; SITETO; MARKETFROM; and ROUNDWOODMARKETFROM. Tables designated SITEFROM specify, for each forest, locations at which processing facilities may be introduced, and whose roundwood resource may be supplied by the indicated forest. Tables designated SITETO specify, for each market, sites from which resources may be shipped. Tables designated FORESTTO specify, for each location at which facilities may be introduced, the forests that may supply that location.

In addition to governing resource flows between processing centres, the costs associated with such flows must also be included. These costs are tabulated in tables designated SHIPTOSINK (ship from forest to roundwood market), SHIPTOSITE (ship from forests to locations of introductions) and SHIPTOMARK (ship from introduction location to lumber and residues market). The contents of these tables provide detail on the cost (per resource unit shipped), of shipping material between the specified destinations.

Processing Facility Operation

The processes involving conversion of roundwood into its conversion products play an important role in the utilisation of roundwood. This importance stems from the returns from the subsequent sale of conversion products and the costs associated with the conversion of roundwood. The

conversion procedures within the test problem were confined to the possible utilisation of roundwood at sawmills of differing capacities. The form of the data for describing conversion activities is now detailed.

The technique used to describe roundwood conversion at a sawmill is known within the forest industry as a mill study. This technique ultimately relies on the determination of proportions specifying breakdowns at each stage of a conversion procedure. Once these breakdown proportions have been established, it is comparatively easy to determine the revenue, cost and profit for the facility, provided that appropriate product revenues, input costs, processing costs and production levels have been identified (Vaney, 1981).

Figure 5.4 shows a conversion procedure for an arbitrary class of roundwood input into a mill. Each such log input can be broken down, using a specified process set, into specified lumber and residue classes. In general, the proportions determined on the arcs within a conversion digraph are regarded as fixed when representing these processes within a mathematical program. In reality, they are random variables, and their specification is dependent upon both the mill under consideration and the sampling procedure used within the mill study.

Test problem data relating the conversion of roundwood (crop-type recoverable volume bucked into log classes) into conversion products (lumber and residue products) is tabulated for each crop-type in the data base. These are the tables designated CONVERT-PRODUCT, the table entries refer to the units of conversion product formed per unit of crop-type log class processed. A description of the conversion products

is given in the table designated PRODUCTSET in the data base.

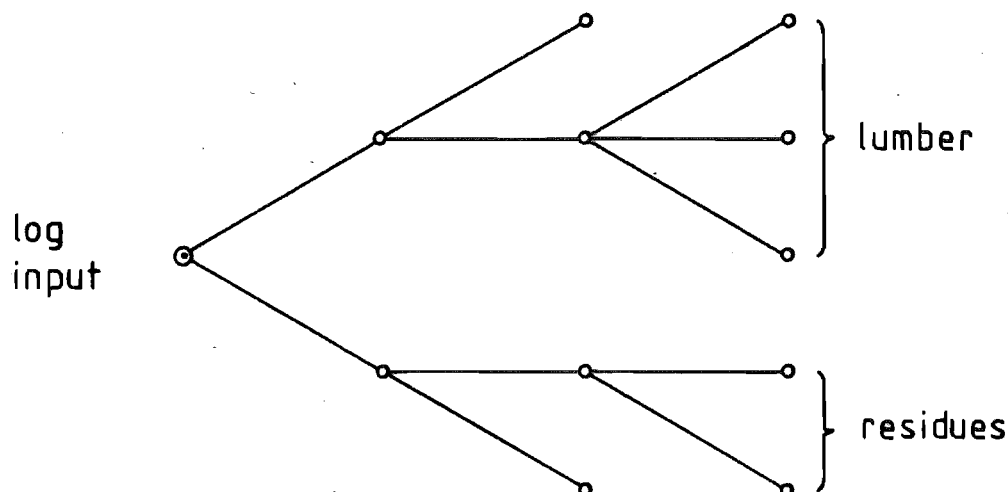


Figure 5.4 : Generalised roundwood conversion subdigraph modelled as a 3-stage conversion process⁴.

The lefthand vertex denotes the resource log input while the set of right hand vertices denotes the distribution of conversion products amongst lumber classes (differentiated by size and quality and residue classes differentiated by type).

The method used to introduce capacity in the test problem was that presented in subsection 3.3.2, the non-interpolating binary mechanism. (Note: the interpolating binary mechanism was used in smaller test problems and was found to function correctly). Data relating to this introduction mechanism are now described.

In order that a facility be coded as part of a model at a specified planning period, the period must be recognised as a possible operating period for the facility. The tables designated OPERATIONPERIOD in the data base tabulate, for each processing facility, possible operating periods.

⁴ A subgraph is used since it permits the illustration of the form, without specifying detail, of the conversion digraph.

Another criterion that may be used to include facilities into a model is that the facility possibly be able to generate a required product set (see section 4.4). This criterion was used in the generating program for the test problem. The required product sets for each facility are the tables designated INCLUDEPRODUCT in the data base. Generators for products in this set (that is crop-type log classes that undergo conversion to a product set inclusive of the required product set - see section 4.3) are tabulated in the tables designated FINDANYCROPLOG in the data base. As outlined in chapter 4, facilities are included within the model structure when there exists a subset of model periods during which the facility may possibly produce its required product set. Furthermore, this subset must consist of adjoining model periods, inclusive of the last.

The tables designated CAPACITYBOUND tabulate for each facility limits that are to be used in imposing the resource allocation constraints (3.12) and (3.13). In this manner lower and upper bound constraints are placed on the roundwood processed by the facility.

The tables designated CAPITALSTREAM tabulate for each facility the payments associated with capacity introduction. Items from these tables are used within the budgeting constraints (3.14) and fixed change mechanism (3.15). The table designated CAPITALBUDGET contains constraint values for the budget constraint system (3.14).

Roundwood, Lumber and Residue Markets

As indicated in section 3.4, stochastic programming techniques could possibly be used in an attempt to realistically represent market structures. However, for the purposes of

the test problem, perfect competition in product markets was assumed. These markets, as indicated in section 3.4, are particularly easy to represent because they exhibit constant returns to scale.

The data required to be tabulated in order to represent these markets are the objective function coefficients, each representing a revenue per unit product transacted at the market in a specified planning period. These coefficients are tabulated in the tables designated SELLPRODUCT and SELLROUNDWOOD. Table SELLPRODUCT specifies the revenue data for lumber and residue products arising from roundwood conversion, while table SELLROUNDWOOD contains revenue data for the disposal of crop-type log classes at roundwood markets. Additionally, data relating to the model periods during which markets exist are tabulated in the tables designated SELLATMARK (SELL AT MARKET), which tabulate, for each lumber and residue market, the model periods during which sales may be transacted. Roundwood markets were assumed to exist in all model periods thus always ensuring feasible roundwood disposals.

5.1.2 Report Writer Output Description

The report writing phase of the problem involves the presentation of results (output from model) to users in a form that is readily understandable. This is facilitated in MODELER (the Burroughs MG System) by writing a program that accesses both the solution results and the data base and constructs reports, using information contained in these files, that are meaningful to model users. This subsection discusses both structural and developmental aspects of the report writing program written for the test problem. It aims to provide an indication of the types of information that may be meaningful

to users. This information is presented in a series of tables, each of which is termed a report. The form and information content of these tables is considered to be descriptive rather than prescriptive. The tables subsequently referenced in Appendix 5.1.2 indicate the form of the reports. All reports generated for the test problem are contained in file RP in Appendix MYFILES.

Current Management Unit Report (1)⁵

The form of this report is indicated in Appendix 5.1.2 p210. It includes for each current management unit (that is, areas established prior to the initial model period), a summary of the management alternatives selected by the model. For each such alternative, the following information is reported:

1. The stocking period prior to the initial model period;
2. Area associated with the management alternative and with the management unit;
3. Cost per unit area over the planning horizon associated with the alternative;
4. The crop-types used in the alternative;
5. An indication whether production thinning has occurred for crop-types in the alternative; and
6. The distribution of both harvested and residual volume by terrain type (viz., hauler and skidder) and time for the alternative.

⁵ The bracketed number(s) after each report name is the page number of a report of this type in file RP.

Future Management Unit Report (8)

The form of this report is shown in Appendix 5.1.2, p211. It is similar to that for current management units except that, on this occasion, the alternatives involve the afforestation of land after the beginning of the planning horizon (that is, they are future management units). Information relating to a land purchase price is tabulated; a single crop-type is used in each alternative for future management units (different alternatives may have different crop-types). Other aspects of the table remain the same as those for current management units.

Crop-Type Age Class Report (10)

The form of this report is shown in Appendix 5.1.2, p212. The essential feature of the management unit reports is that they are stand level reports, whereas crop-type/age class reports present information at a forest level (c.f., section 2.2). The tabulation shows the distribution of both

1. harvest volumes and areas, and
2. residual volumes and areas

over time, by terrain type, for each crop-type/age class.

Crop-Type Log Class Report (17)

The form of the tabulation is indicated in Appendix 5.1.2, p213. The report is similar to the crop-type/age class report, except that instead of detailing information relating volumes in age classes, it details information relating volumes to log classes. Specifically, the distribution of harvest and residual volume, by log class, over time, for each crop-type is given.

Residual Volume Report (19)

This is an aggregated forest level report. The form of the tabulation is specified in Appendix 5.1.2, p214. The distribution of residual volume over time is given for each forest. This can be envisaged as either the aggregation of crop-type/age class residual volumes over terrain types, age classes, and crop-types, or aggregation of crop-type log class residual volumes over log classes and crop-types.

Harvest Volume Report (20)

The harvest volume report is identical to the residual volume report, except that the word "residual" is replaced by "harvest". The form of the tabulation is indicated in Appendix 5.1.2 p215.

Forest Arc Reports (21), (22)

These are reports on transportation arcs from forests (the source) to either roundwood markets or sites where roundwood conversion takes place (the destination) as depicted in Figure 5.1. For the sake of clarity, a separate report is generated by the RW program for each source/destination pair. The form of the tabulation is shown in Appendix 5.1.2 p216. These reports provide details for each crop-type on the volume by log classes which are shipped along the arcs over time.

Location Operation Report (25)

Reports of this type indicate for a specified location, processing facilities that are introduced at that location during the planning horizon. An example of the tabulation is

shown in Appendix 5.1.2, p217. A horizontal bar graph is used to show the times that processing facilities are operational. These times will always be an interval starting with the introduction and finishing at the end of the planning horizon. In the report, comparison of the bar graph with the time scale immediately above it gives the desired interpretation of operating times.

Site Capital Finance Report (26)

These reports are best understood in conjunction with the associated Site Operation Report, in that they document the payment scheme associated with introductions of a processing facility (see subsection 3.3.1). A statement of the initial amount required to effect the introduction is given, along with the repayments made during the model time frame. The form of the tabulation is illustrated in Appendix 5.1.2, p218.

Facility Input Report (27)

Generally, reports detailing information about the operation of processing facilities need to provide details on both quantities and timing of resources input and output. Facility input reports are to be seen as part of a plant operation report. The tabulation form in Appendix 5.1.2, p219, designed for a sawmill, shows the distribution of roundwood volume by crop-type and log type used as input to the mill.

Facility Output Report (28)

Facility output reports are to be considered as part of a facility operation report. The form of tabulation

presented in Appendix 5.1.2, p220 lists for each product, the distribution of product units by time assuming the facility is operational. These tabulations are aggregated from all processes giving rise to the indicated product.

Generally, facility operation reports should include process-related information as well as the aggregate input and output reports. Thus processing centres within processing facilities could be reported on.

Location Arc Reports (30)

The arcs reported on are transportation arcs that begin at a location where facilities that utilise roundwood may be operational, and end at destinations where roundwood conversion products may be sold (see Figure 5.1). The form of the tabulation is illustrated in Appendix 5.1.2, p221. Details tabulated for each product include a description of the product; the number of product units shipped along the arc over time; and transportation costs in dollars per unit product shipped.

Market Product Report (23), (31)

As indicated in section 3.4, marketing mechanisms play a central role in forest utilisation management problems. It is for this reason that attention should be paid to detailing, during the report writing phase, the marketing assumptions adopted as well as the marketing strategies indentified. The following tabulations are suggested for each market:

1. The distribution of product units sold, by identity and amount, over time;
2. The distribution of product revenues gained, by product, over time;

3. The probabilistic nature of market structures if stochastic market representations are employed.
4. The specification of product bounds should they be imposed to reflect market structure; and
5. Any related information pertaining to market structure.

Examples illustrating 1 above are given in Appendix 5.1.2, pp 222 and 223 for roundwood, and lumber and residue markets respectively.

5.1.3 Singularity Report

Application of the branch and bound procedure for the test problem was marred by the reported occurrence of singularity in the basis matrix. A discussion of the possible reasons for this phenomenon which, conceptually, should not occur, forms the basis for this subsection.

The two reported singularities arose during the TEMPO procedure MXINT (viz., the branch and bound procedure, which requires an LP solution to the relaxed integer problem before being initiated). The action taken by TEMPO was to initiate an auto-recovery routine that removed a vector from the basis and allowed computation to continue subsequent to constructing a new basis.

Conceptually, this problem should not arise in that the condition necessary for singularity, a zero pivot, is always avoided during pivot operations (this is shown in Appendix 5.1.3). However, finite machine arithmetic and an ill-conditioned basis could give rise to such a situation.

The method by which the basis inverse is stored can influence the precision of machine arithmetic when computing the basis inverse. TEMPO uses the product form of the

of the inverse, and thus requires the storage of a sequence of Eta vectors (see Appendix 5.1.3); machine precision in calculation of the inverse falls off as the number of Eta vectors grows.

Further, the occurrence of an ill-conditioned basis in linear programming can be a transitory phenomenon, since at each iteration a square system of equations is being solved and subsequent iterations may involve different bases (depending on the LP solution code) that are no longer ill-conditioned. This is of relevance to the branch and bound procedure since it operates by relaxing integer restrictions and solving, in this case, an associated linear program.

After solving the test problem (the solution gained is known to be optimal because of the test problem construction), the solution was verified by forcing integer variables and solving the resulting LPs. A further investigation was undertaken to determine if, by reducing the number of iterations between re-inversion of the basis matrix, the reported singularities could be removed. This would reduce roundoff error since it would limit the number of Eta vectors that could be used in the product form of the inverse. The default number of iterations between re-inversion in TEMPO is 50. A demand implementation of the re-inversion routine with the default value changed to 25 was used to solve the test problem. However, the reported singularities remained.

Subsequent to these runs, no further examination of the reported singularities was made, largely because of time constraints. Further investigation would have to consider whether or not the problem arose because of the structure of the constraint matrix. In particular scaling

of the constraints used to regulate capacity, systems (3.12) and (3.13) in subsection 3.3.2, may be required since these constraints contain coefficients (r_{is} , r_{is}^*) whose magnitude is large in relation to other entries in the constraint matrix. A possibility that must also be considered was the implementation and functioning of the TEMPO routine MXINT at Canterbury University during the time of these program runs. The TEMPO system is no longer supported by Burroughs Corporation. Before its withdrawal of support, Burroughs recalled the routines used in solving decomposed problems, presumably because of software errors. The routines associated with the revised simplex method (PRIMAL) and dual simplex method (DUAL) are considered amongst users at Canterbury to work well. However, the same confidence is not held of MXINT.

In view of the robustness of branch and bound methods to solve Mixed Integer Linear Programs with small numbers of integer variables, the problem encountered is not considered to limit application of mixed integer programming techniques to FUMPs. Singularities arising from the ill-conditioning of a basis may possibly be removed by redefining the structure of the problem or updating the inversion frequency of the inverse. Structure and inversion frequency, then, constitute the means by which a modeller can combat this phenomenon. Changes in inversion method and inverse storage method usually cannot be as readily made as changes in model structure or changes to inverse "house-keeping".

5.2 MATRIX GENERATION DEVELOPMENT AND VALIDATION

This section considers ways in which Matrix Generator (MG) programs are currently developed and validated. It examines the methods of validation documented by Fourer (1983) and, in light of constructing the MG program for the test problem, reasons that even with these aids, caution must still be exercised during a validation phase. It is also recommended that MG programs be developed and validated concurrently.

Specifying input to a mathematical programming system on a constraint by constraint basis (row-wise specification) or variable by variable basis (column-wise specification) would be extremely time consuming for large-scale linear systems. Moreover, the possibility of introducing errors is high. For these reasons, specially designed computer programs called matrix generators are used. Generally, MGs rely on models possessing structural attributes, which result in the repetition of certain classes of constraints or variables. These structural features are exploited by generators. Williams (1978) indicates that the use of MGs allows attention to be focused on the structural aspects of the model while the repetitive aspects are automated.

It was largely for the above reasons and also for a desire to gain experience with MG systems that a decision to write the MG program used for generation of the test problem was made. A full program listing is given in file MG in Appendix MYFILES.

Practical methods used to validate matrix generators have been documented by Fourer (1983). These are set out below:

1. A matrix generator program may be double-checked by someone likely to recognise obvious inconsistencies with the problem to be solved;
2. A matrix generator program may be executed, which may signal certain errors in compilation or execution;
3. Output from a matrix generator program may be examined or tested by specifically designed computer routines (such as software to write the constraints for the problem);
4. Output from a matrix generator may be input to the solution algorithm, where errors may be reflected in infeasibility, unboundedness, or implausibility of an optimal solution.

Although this list details practical methods of validation, it does not constitute licence to write MG programs that will generate significant amounts of output and then proceed to test such programs. To do this may result in generating and solving the wrong problem.

Solution of the wrong problem could arise because each of the validation methods listed is no guarantee against trapping errors. Method 1 assumes the person checking is familiar with the structure of the problem and has enough patience to search for inconsistencies. Method 2 relies to a large degree on the interpreter or compiler that executes the MG program, which in itself may be subject to errors. An argument could be made that the retention of software errors in a system occur in inverse proportion to the use of that system (fewer users imply more errors remain undetected). Mathematical programming systems do not generally have large numbers of users, and there is a tendency, therefore, for

software errors to remain embedded in these systems for a considerable time. The output for method 3 may be of sufficient volume to make validation difficult. Use of 4 above should be reserved as a final method of validation because, as far as is possible, output from MGs should be validated prior to its use by a solution algorithm.

One application of method 3 that was used to good effect in the development of the MG program written was to examine the MG output either manually or through use of a program called EQUATIONWRITER (available to users of the MODELER/TEMPO system at Canterbury University) to list the constraints of the MG output. The amount of generator output, was restricted, either through use of the database or else through using individual program fragments from the MG program, and possibly a combination of these methods. The approach was found to be particularly convenient as a validation mechanism.

Seemingly, a more sensible alternative to writing and subsequently testing large MG programs would be for the development and validation to be undertaken concurrently so that, by the time the MG programs have been completely written, its component parts are known to function correctly.

While developing MG programs, one should have as much control as possible over the MG program during execution. In this way, the attributing of cause to a fault can be simplified. Some measure of control can be gained by incorporating basic debugging features within generating programs; further control measures may incorporate placing error traps in generating programs. If additional software is used to validate a model, then some measure of control

will be lost if this software has errors, and the determination of a causative agent for identified errors can become confusing.

The approach suggested in developing and validating MG programs is to use jointly the validation methods detailed by Fourer (*op. cit.*). Modular construction and testing of small individual program fragments should be adopted, and subsequently, linking of program fragments can then be checked.

5.2.1 Observations on Matrix Generators

This subsection contains an examination of matrix generating procedures largely from an historical viewpoint, discusses some of their inconveniences, and suggests and presents ways to overcome these.

Fourer (1983) in his paper entitled "Modeling languages versus matrix generators for linear programming" outlines difficulties associated with traditional matrix generating methods and proposes the use of Modeling Languages (MLs) for the specification of Linear Programming problems (LPs) to a solution algorithm and also for subsequent report writing purposes. Fourer's proposals are relevant not only to LPs but also to any situation involving the generation of large scale matrices. This suggestion forms the basis of the discussion in this subsection. Also mentioned is a method employed by Garcia (1984) to generate Forest Management Problems. Fourer (*op. cit.*) initially identifies two forms of linear program.

1. The modeller's form, and
2. The algorithm form.

The first is the usual method to describe LPs, that is, algebraic notation involving subscripting with summation over various index sets to specify the model structure. Alternatively, the algorithm form is machine-dependent and consists of data structures that facilitate storage of the problem and execution of the solution algorithm. The modeller's form is described as being symbolic; concise; and understandable. The symbolism arises because the structure of the problem is presented using symbols, conciseness because the problem description is almost as brief as possible, and understandable in that the problem is easily read and is intelligible to a reader. This is contrasted to the algorithm form, which is described as being explicit rather than symbolic since actual data values are specified in the algorithm form. It is convenient rather than understandable in that it comprises structures that are machine-related.

Both the modeller's and the algorithm form are necessary because problems are conceived in modeller's form and presented to the machine for solution in algorithmic form. No single form of expression could serve the purposes of both forms. Thus, a conversion, termed a "translation" between the two forms is required.

It is how this translation should be undertaken that constitutes Fourer's proposal. The use of a special ML to perform the task is suggested. Such MLs are envisaged as being more closely oriented towards the user (that is, the modeller's form) than are current MGs, which incorporate in their design features that are closer to the algorithm form. The use of MLs would enable the user to specify the problem in a manner closely resembling the algebraic notation of the modeller's form rather than the admixtures of terminology

from linear algebra and computer science that are used by current MGs for problem specification.

Fourer (*op. cit.*) provides further criticism of MGs, namely, representation of data within the data base; the naming of LP components; the ordering of coefficients; and the representation of special constraints. Data representation is considered simplistic in that most MGs allow only one- or two-dimensional structures in which to store numerical data items. The naming of LP components is considered restrictive in that most algorithmic forms employ a standard form for data representation known as an MPS (Mathematical Programming System) form, in which problem data is specified by use of ROW, COLUMN, RHS, RANGE and BOUNDS sections. This form permits a maximum of eight character names, which may give rise to restrictions associated with the modeller's form, especially if either large or numerous sets are to be subscripted. The ordering of coefficients refers to the order in which the matrix is generated. Generally, column-wise generation procedures work faster. This is an anachronistic feature that is strongly influenced by the MPS form in which the matrix is specified by column. Finally, the representation of special constraints requires special sections and markers within the MPS form, which is considered to be an irrelevance to the modeller.

The order of generation has a marked effect on generation times within the BURROUGHS MODELER System. Modeler is unusual in that generation using ROW statements (not necessarily a row-wise generation procedure in that repeated use of ROW statements can be made for each column in a column-wise generation procedure) is several orders of magnitude faster than that using COLUMN statements. Data

contained in Appendix 5.2.1 illustrates this. Differing generation times are generally linked to storage methods within internal data structures, and faster access times for ROW or COLUMN statements to cells within these data structures. Because most matrices for reasonably large scale LPs are sparse, choosing a doubly-linked data structure to represent matrix elements would seem a logical starting point in the construction of a MG system. Average search times could then be expected to decrease, and discrepancies between row and column generation times should be abolished.

The differing execution times for ROW and COLUMN statements within MODELER (see Appendix 5.2.1) formed the basis for undertaking revisions of the MG program used to generate the test problem. These revisions undertaken during program development ensured that as much of the generation procedure (column-wise) as possible was conducted using ROW statements.

The system adopted by Garcia (1984) to generate Forest Management Problems, the structural constraints for which were outlined in subsection 2.2.4, includes an interpreter that accepts commands that are user oriented, these commands being acronyms derived from terms employed by foresters to describe the corresponding situation. Information supplied as parameters to these commands is subsequently used in the formulation of the problem. This approach can be considered as a step towards being user friendly. It shows a similarity to Fourer's proposals, in that the notation employed is that of the modeller. Such an approach is contingent upon the writing of an interpreter program in addition to the generating program.

Garcia's method, however, seems preferable to allowing users to have direct access to the problem data base or

generating program, which would involve confrontation with data or program structures that are of little relevance to a user, and moreover are potentially confusing, being more closely linked to the algorithm form than the modeller's form.

Currently specialised languages used for MG purposes are undergoing a hiatus in their development, and until such time as full modelling languages emerge, MG programs could be written in high-level languages such as Algol, Pascal, or C. In this way, some of the adverse effects of current MG languages could be avoided.

5.2.2 Observations on Report Writers

As indicated in subsection 5.1.2, the report writing phase involves the presentation of output in a form that is intelligible to users. The form in which output is presented is often the sole criterion by which users judge mathematical programming techniques (Williams, 1978). This subsection examines the report writing phase and notes that modularity in construction and use of information contained within the solution may be used to facilitate the execution of Report Writers (RWs).

The report writer program developed in this study, file RW in Appendix MYFILES, is largely modular in construction, each module corresponding to the formation of an output report. Within each module, row and column names are constructed using various MODELLER control structures and then checked for activity in the optimal solution. If non-zero activity is detected then information is included in the appropriate report.

Information contained in the model solution can be used to advantage during the report writing phase. For example, if a binary variable representing capacity introduction at a processing facility has value zero, then all related processing activities must also be zero. The incorporation of FREE constraints (that is, linear combinations of selected sets of variables that do not enter the model as a constraint) may also be used to advantage during report writing. Inclusion of features such as these have a marked effect on decreasing execution times for RWs.

One feature that proved to be distracting during the development of the report writer was that references to the model structure accessing technological coefficients are not directly allowed by the MODELER report writing system. Such references are allowed only during the generation phase when the model structure is being formed. This necessitates the reconstruction of technological coefficients from the data base should they be required during the report writing phase. Unfortunately this involves repetition of work that has already been performed during the generation phase.

Normally, RWs should execute faster than MGs because, given the approach that values of some activities can supply information relating to the value of others, the activity of every row or column need not be examined. Furthermore, RWs do not normally require the generation of technological coefficients as do MGs.

Fourer (1983) indicates that RW and MG programs suffer from the problems of validation, modification and documentation, but that these are more severe for MGs than for RWs, largely because output from RWs is easier to understand. This

observation is confirmed in this study because the validation of the RW program was found to be markedly easier than that of the MG program. A large part of validating an RW program may be done by the (binary) operations of addition and subtraction, comparing the solution output with that reported by the RW. Validating an MG program, on the other hand, requires checking both numerical (model coefficients) and alphanumeric (row names, column names, etc.) quantities.

As a result of the computational experience gained from this study and in consideration of the state of development of matrix generating and report writing systems, it is suggested that the development of generating and report writing systems should include features that allow for more flexible definition during the generation phase, and selective reporting during the report-writing phase. Existing high-level languages could be used in the development of generating and report writing systems for the classes of mathematical programs specified by FMPs or FUMPs.

5.3 RESTRICTED ANSWERS FROM PLANNING MODELS

Modelling, by providing information, is a framework used to assist planners in their role as decisionmakers. The techniques available to do this have greatly improved since introduction of modelling methods in the 1940s. Nevertheless, planners still often find themselves in situations where answers obtained from planning models are restricted in terms of both information content and accuracy.

Most planners recognise that restricted answers are associated with information loss. Modellers may make many decisions during the course of model development that determine the form of the information that planners will receive. These decisions are not necessarily in the best interests of planners, and an argument can be presented to illustrate the conflict of interest between planners and modellers. Their respective roles as users and providers of information are not necessarily complementary. Planners seek reliable information at reasonable cost, while modellers are left to resolve the inherent conflict of cost and reliability. The inherent tension arises because increased reliability translates to additional expenditure on model development and data collection, this can, to some degree, be resolved by determining what information is to be provided at what cost.

This section identifies some of the means by which inaccuracy through information loss can occur. The approach cannot be specifically remedial in that each planning situation is different, but instead is largely descriptive, outlining how such situations may arise. Situations in which information loss or inaccuracy occur lead to the optimum-seeking behaviour of optimising models rather than to the identification of optimal solutions as purported.

5.3.1 Model Aggregation

Aggregation may arise in various ways during model construction. Modellers may even deliberately use aggregated forms within a model structure. Motivation for doing so may come from a desire to reduce the size of the problem, a desire

to facilitate application of a specific solution technique, cost considerations, or inability to estimate data items.

Regardless of how aggregation is introduced into a model, it becomes a means of information loss. Furthermore because most forest planning models are solved by deterministic techniques that have no regard for the precision and accuracy of data estimates (all data items are treated as though they are exact), any unwise aggregation can only exacerbate the introduction of errors into a model solution.

Aggregation in planning models may lead to difficulty in solution implementation. Rose (1984) considers this problem to arise in Forest Management Problems, in that the preparation of a management schedule allows great leeway in implementing activities within an aggregate. To illustrate this point, consider the area within an aggregated management unit. Difficulties arise if proportionate areas are to be considered. Such areas must still be capable of being physically located on the ground.

Perhaps the most disturbing aspect of aggregation is that, in a model component, it can affect a solution in such a way that the ramifications are not confined to the aggregated model component. That is, misrepresentation in part of a model can affect representation of answers in other parts of the model.

5.3.2 Estimation Errors

Estimation errors may affect the precision and accuracy of data items used to specify a model. Moreover, these errors are likely to differ in magnitude between model components, because data collection procedures may differ in both method and the environment in which data are trapped.

For example, in forest planning models, data relating to forest regulation are likely to be more precise and accurate than those concerned with utilisation and marketing.

In multi-period planning models time also adds to estimation errors of data items in future model periods. Statistical theory indicates these estimates will be imprecise. Furthermore, the greater the time interval over which estimations are made, the more imprecise these estimates will be.

Note that the heuristic proposed by Hoganson and Rose (1984) to solve Forest Management Problems acknowledge the presence of estimation errors as a motivating factor for searching for their new solution technique. The solution mechanism proposed (see subsection 2.4.4) relaxes primal feasibility. The rationale given is that the values used to impose feasibility are themselves subject to error.

5.3.3 Stochastic Elements

Stochastic elements occur within forest planning models, which are in addition to the errors of estimation associated with data items. For example, market prices and transportation costs are, in fact, values from suitably defined random variables. Under such situations, estimation errors are compounded by the fact that the scalar used is a value from a random variable.

The precision of a large number of data items associated with forest planning models remains unknown. Roundwood volume estimates provide a suitable illustration - these may be obtained by the projection of a state vector of growth parameters over time (Garcia, 1981). No information concerning

precision of estimation is projected. Presumably, if such estimates were formed, the errors attached to these would not be small.

Unfortunately, some of the most important data items in forest planning models are in fact random variables, and these have to be characterised by a scalar for solution purposes. An obvious choice for a scalar would be the expectation of the random variable. However, estimating expectations greatly enhances the problem of data collection, and understandably the procedure so often adopted is that any "reasonable" value will do.

The stochastic nature of forest planning problems can sometimes be reduced by the inclusion of additional constraints and variables into the model (thereby lowering the level of aggregation) so as to further discretise continuous distributions. However, when this is undertaken, the number of data items required to be estimated increases concomitantly.

5.3.4 Cost Considerations

Cost expenditure during modelling, besides influencing the level of aggregation, may also hinder the development of adequate planning schedules. Computational experience with Forest Management Problems indicates that solution costs may be excessive (Rose, 1984).

The costs of preparing an adequate planning schedule include the number of model runs required to produce the initial schedule, the runs required to update the schedule as time proceeds, and the preparation and maintenance of the data base. Of these, costs associated with the data base may

constitute the largest outlay. Reductions in data base cost could be made by considering the size of the problem in relation to aggregation of model components, and the advantages of writing computer software that may aid in the preparation and maintenance of the data base.

Solution costs depend on solution techniques adopted. Either heuristics or solution mechanisms that account for structural elements of the problem (inclusive of network codes) offer the best means to reduce solution costs for Forest Utilisation Management Problems.

Most costs in developing planning models are initial costs, and subsequent change to the structure of a model can usually be achieved economically, provided that the model has been adequately maintained and suitable documentation exists. These costs are likely to be strongly influenced by the experience of the modeller with the class of model being developed.

5.3.5 Uncertainty in Economic Efficiency Measures

When money (strictly, some function of it) is chosen as a measure by which to judge economic efficiency, then each possible measure formulated will allow ordinal classification of efficiencies. However, there is no guarantee that different measures will preserve the order of classification (c.f., section 2.1).

The use of money as an economic efficiency measure is subject to

1. changes in the value of money over time ("money being an ordinal measure in unstable units", Leslie, A. 1984, pers. comm.), and

2. commodity exchanges being made at varying prices over time.

In forest planning models, the first is usually ignored; in response to the latter, the assumption is usually made that all costs and revenues increase or decrease in constant proportion to each other. (Note - 1. and 2. are not strictly independent since changes in the value of money influence prices). The validity of this assumption in the long term is questionable in that the relativities of costs and revenues can be altered. Certainly the introduction of new technologies offers one possible means by which relativities may change.

Under circumstances in which these relativities are not preserved, economic efficiency measures have associated with them a high degree of uncertainty.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

This chapter contains both a discussion and conclusion to the study undertaken. The discussion, contained in section 6.1, centres around the following facets of Forest Utilisation Management Problems; regulatory aspects of resource flows, matrix generation and report writing, and model formulations and solution codes. The conclusions, contained in section 6.2, present findings of this study in relation to the study objectives outlined in chapter 1.

6.1 DISCUSSION

Regulatory Aspects

The approach to Forest Utilisation Management Problems adopted in this study has been to concentrate on the means by which resource flows between production centres comprising forests, processing facilities, and markets, can be regulated. This allows integration of activities between processing centres and the search for efficient means of integration to be undertaken. These analyses, however, are only part of the development work needed to be undertaken in order to represent Forest Utilisation Management problems as a class of mathematical program. Absent are detailed production models for various types of processing facilities. Modelling types of processing facilities in detail is a topic beyond the scope of a single study such as this, but it is envisaged such facilities could generally be represented as multistage processes. Attention was paid in chapter 4 to a general aspect of such processes, where it was shown that the mapping

of resource inputs to products output can be used in problem generation.

Special attention was paid to the regulation of roundwood flows from forests. Alternative techniques for regulating roundwood removals were discussed in section 2.2. These techniques, described as model I or model II were shown to provide differing interpretations of management unit areas (Johnson and Scheurman, 1977). Neither form of representation is the better in all circumstances. Model I formulations allow management activities for a management unit to be specified in greater detail thus permitting the possibility of more realistic representations of these activities; however, the number of variables required to formulate model I representations quickly increases in response to scale of the problem. Implicit in the choice between these two formulations is the question of aggregation. Model II formulations may be more highly aggregated because the underlying structural differences in area regulation can affect the specification of management activities within management alternatives. That model II formulations generally require fewer variables than do Model I is understandable from a consideration of the relative level of aggregation.

Problems associated with roundwood resource flows were discussed in section 3.1, where the necessity for adequate log grading systems by which roundwood flows may be defined and measured was identified (Whiteside and Manley, 1985). The importance of this in relation to FUMPs cannot be underestimated, because the accuracy of representation of not only roundwood flows, but also flows subsequently arising from processed roundwood are dependent upon such grading systems. From a modelling viewpoint, log grading systems

offer the facility to reduce stochastic elements within a model structure by way of increased accuracy in representing wood supplies.

Regulation of resource flows to and from processing facilities was discussed in chapter 3. Amongst the material presented was the use of integer programming techniques to introduce processing capacity at facilities and resolve the problems attendant with introduction (section 3.3). Foremost in the decision to use integer programming techniques to resolve these problems are questions concerning reality in representation. Since the use of linear programming to resolve these problems necessarily implies assumptions be made about representation of capacity at processing facilities. For example, the requirement that facility capacity is continuous in an interval from zero up to the maximum capacity is probably unrealistic for large-scale facilities because an observable trend amongst utilisers of roundwood resources is that, the larger the roundwood input required for operation of a facility, the more likely that the full capacity of the facility will be established at the time of its construction (discrete increments of capacity).

Thus, the approach using integer programming is considered to be of value in that the ability to incorporate even the small number of integer variables required for processing capacity introduction gives worthy returns in representational accuracy. However, the cost that must be paid is the increased computational burden associated with programs containing integer variables. Nevertheless, existing algorithms for solution of these programs possess practical limits as to how many integer variables may be included

(dependent on the algorithm). The number of integer variables required to specify capacity additions within even a large Forest Utilisation Management Problem would generally be less than 100, and thus be within limits of existing algorithms.

Errors manifest in the modelling process have been identified and discussed in section 5.3. These give rise to representational and solution errors for FMPs and FUMPs. Exact magnitudes of representational errors are generally unknown, but a statement concerning the relative magnitudes expected for different types of data can be made. Those data relating to forest management activities are often both more accurate and precise than those concerned with subsequent utilisation of roundwood. The acquisition of adequate data, so often influenced by the activities of interested parties, is a factor to be considered in application of optimisation based techniques to forest planning problems. Any optimisation model must necessarily include data from all planning scenarios being examined, whereas simulation techniques can operate by detailing the activities of selected scenarios (better scenarios may exist) thereby allowing a greater concentration of effort on data acquisition.

Matrix Generation and Report Writing

In chapter 5 matrix generation and report writing procedures were discussed in relation to their development and maintenance. The applicability of the validation methods documented by Fourer (1983) were discussed and the recommendation made that Matrix Generator programs (MGs) be developed and validated concurrently using these methods. A further recommendation was that, in so far as possible, development

of MGs should be modular with additional testing conducted at the module linking phase (section 5.2). This approach differs from usual applications of high-level computer languages in that program modules (fragments, sections) written before validation are envisaged to be relatively smaller in size.

In subsection 5.2.1 it was indicated that matrix generation necessarily involves a conversion between a modeller's form and an algorithm form. Problems are conceived in modeller's form and presented for solution in algorithm form. Fourer (1983) proposes special-purpose Modelling Languages (MLs) to translate between these two forms. Specialised languages used for MG purposes are currently undergoing a hiatus in their development; presently, these languages contain features that hinder clarification in the modeller's task. It was recommended, therefore, that high-level languages having facility both to manipulate numerical expressions, and items within sets using set operations, be used for generation purposes.

An important part of this study, not yet explicitly stated, has been developing the capability to manipulate items within, and to establish relationships between, elements from finite sets. This capability permits ease in formulation and generation of problems, and may also serve an investigative role during these problem-solving phases and also during report writing.

For example, the entire subject matter of chapter 4 is reliant upon establishing relations between certain resource sets. These relationships were established using material from the theories of sets, functions, and digraphs. In this

case the importance of these relations was that they permit the prevention of redundancy in some situations. As indicated, this could occur when a FUMP contains multi-stage processing facilities.

Similarly, digraphs were used in chapter 2 to describe the differences in model I and model II forest regulation procedures. These digraphs were used to illustrate the different crop-type/age-class area partitions induced for management alternatives having the same harvest/reforestation options (section 2.2). These digraphs can also be used as the basis for generating procedures to specify management alternatives (see subsection 5.1.1).

Thus, the theories of sets, functions, and digraphs in finite mathematics serve as aids during the FUMP phases of formulation, generation, and report writing.

Ultimately, modelling languages used for generating and report writing purposes must aim for both user intelligibility and speed of operation. Currently good working compromises can be gained through using high-level languages to construct interpreter-generator systems (Garcia, 1984). In order to induce intelligibility, sacrifices in operation speed are accepted.

Model Formulations and Solution Codes

The recent experience of Hoganson and Rose (1984) indicates that their choice of solution code for FMPs was influenced by both the formulation of the problem and the imprecision of the data items used in the problem specification. Although the solution technique adopted is strictly a

heuristic based on linear programming and duality theory (subsection 2.4.4), it is of interest in that its development is indicative of a search for a *better* planning model.

The question of what constitutes a better planning model necessitates a comparison of how planning models are formulated and solved. Comparisons of solution mechanisms are perhaps more readily resolved than those concerning model formulations. Recognition of this difference is important, and is indicative of the inevitability of involving subjective elements in modelling. Model formulations that are better imply that criteria exist by which ordinal classifications of utility can be made. Although such criteria do exist, they are by no means sufficient to allow such classification in all circumstances (criteria that permit orderings in this way are known as preference relations (Intriligator, 1971)). Thus, there are no well-defined criteria to facilitate choices amongst model formulations, and modellers are free to employ their own ordering (preference) relations in the selection of better formulations. The manner in which these preferences are made is likely to be subjective depending on modelling experience. This observation explains to a large extent the plethora of modelling techniques used to address similar problems. Note too, that one purpose of discussion is to modify and form preferences. On the other hand solution mechanisms (assuming alternative methods provide the same answers) may be more readily compared using such criteria as the cost of computing resources, (that is, the costs of machine memory, and processing time, etc.).

The development of heuristics to solve forest planning problems besides indicating a quest for better planning models are interesting from a solution viewpoint. Those of relevance to this study have, to date, been concerned with the solution

of Forest Management Problems (FMPs), and can be regarded as derivatives of mathematical programming problems in which; either

1. primal feasibility is relaxed; or
2. primal optimality is relaxed.

An example of each of these methods is briefly discussed below. Computationally, these, or similar techniques, may offer significant advantage in that they permit application of facile solution techniques.

An example of a heuristic illustrating relaxation of primal feasibility is that developed by Hoganson and Rose (*op.cit.*). This model operates with an objective where production costs are minimised. The extension to allow maximising of net returns is an urgent need. This form of heuristic is probably best suited to providing answers to FMPs. To extend the approach to FUMPs would require estimation of more than one class of dual variable, but this implies knowledge of the economic behaviour of those variables, which is at best imperfect.

Models formed by the relaxation of primal optimality take account of solutions that are feasible and close to optimality. The motivation for using such heuristics is that computational experience with FMPs indicates that convergence is almost asymptotic as optimality is reached (see Littschwager and Tchong, 1967 and also subsection 2.4.2). Implementing relaxation of primal optimality requires the adoption of suitable termination criteria. If the programs being solved are linear, such criteria are easy to construct, the value of the dual variables being available at each iteration. Thus the values of primal and dual objective functions could be used to construct absolute or relative

termination criteria (Gass, 1975). Such criteria can also be combined with solution mechanisms that take into account structural aspects of the problem. For example, the application by Littschwager and Tcheng (*op. cit.*) of a relative termination criterion in conjunction with decomposition for solving FMPs.

The development of heuristics to solve forest planning problems, especially FMPs and FUMPs, will probably be an area rich for future research and debate. The motivation, or possibly temptation, being to find "reasonable" solutions to these problems without the computational burden of mathematical programming. As previously indicated, there are no quantitative measures that can be used in all circumstances to assess the relative merits of such methods. However, in substantiating an argument for use of these methods recourse must quickly be made to the qualitative and subjective aspects of modelling (Rose, 1984).

Mathematical programming will probably play an increasing role in answering questions stemming from FMPs and FUMPs. This role may at times appear to be not clearly defined, which is considered to be more a feature of the planning process *per se*, than the applicability of the technique. Increased use of these techniques is envisaged to arise from

1. the desire to recognise FMPs and FUMPs as mathematical programs,
2. an increased ability to solve the problems that arise.

The scale at which FMPs and FUMPs are to be solved will be determined through interaction of conflicting factors.

Leading to scale increases for FMPs are the needs for both a wider range of management alternatives and simply

bigger problems. While for FUMPs, scale increases would stem from a wider range of forest management, transport and utilisation and marketing alternatives, being given consideration.

Opposing this increase in scale are both the modelling of situations where representational errors render the results questionable (see section 5.3), and costs associated with data base formation and maintenance.

Accompanying any increase in scale will be a concomitant need for efficient solution techniques. Generally, such large-scale problems will require the solution of one or more LPs. Even allowing for refinements to simplex codes such as compact inverse methods and sparse matrix techniques, solution of these LPs could still be an expensive operation.

The problem of scale in relation to FMPs is well illustrated by the forest planning system FORPLAN developed for the United States Department of Agriculture (USDA)*. This system is currently under criticism from several authors (Rose, 1984; Apple, 1982; Walker 1982 cited in Rose, *op. cit.*; Berck and Bible, 1984) for both excessive solution costs and the inaccuracy of its results. The criticism regarding solution costs stems both from the costs associated with data collection and data base maintenance and use of the revised simplex method for the solution of large-scale problems.

The scale problem has motivated the search for solution techniques other than traditional LP techniques. Decomposition methods have been proposed for FMPs with model I area regulation (Littschwager and Tchong, 1960) and model II (Berck and Bible, 1984; Garcia, 1984) formulations. These are standard Dantzig-Wolfe decompositions. In each case,

*(Johnson, Jones and Kent, 1980)

the linear programs arising at the level of the subproblem have facile solutions.

To propose solution techniques for FUMPs, a distinction must be made between those models with integer variables and those without. The presence of integer variables is accompanied by increased computational effort. Given a linear program, network formulations of the problem (processing networks) along with network solution codes potentially offer an attractive means of solution (subsection 3.5.2). For large-scale problems involving integer variables, either Bender's procedure or else a branch and bound method with an embedded processing network code could be used. For qualitative reasons, I favour the latter approach.

6.2 STUDY CONCLUSIONS AND RECOMMENDATIONS

From the research undertaken and reported here, it is possible to draw the following main conclusions as they relate to the four study objectives that were outlined in chapter 1.

- 1.(a) Resource flows in Forest Utilisation Management
Problems can be modelled by mathematical programming techniques and are amenable to implementations of linear or mixed integer linear programs.
- (b) Problems that arise in consideration of flows to and from processing facilities, the capacity of which is being expanded (or contracted), can be resolved using mixed integer linear programming techniques.
- (c) Certain flows within Forest Utilization Management Problems are characterised by the presence of stochastic elements; for example,

- (i) resource flows from forests, and
- (ii) resource flows at markets.

In such situations it is recommended that consideration be given to reducing stochastic effects either by representing the distributions concerned discretely (representation of the distribution, presumed continuous, by a countable set of real numbers), or by application of stochastic programming techniques. Possibly a combination of these may also be used.

- 2.(a) Many existing matrix generating techniques employ admixtures of "terminology" from computer programming and matrix algebra that have no relevance to the task being performed.
- (b) Special-purpose generators written in languages that include syntactic expressions for basic set operations (such as: is contained in; is not contained in; union; intersection) along with the facility of manipulate numerical and alphanumeric quantities may offer viable alternatives to existing matrix generating languages.
- (c) Certain topics from finite mathematics, namely set theory, mapping theory, and digraphs can be used as aids during model formulation and generation, as shown in chapters 2, 3 and 4.
- (d) Although this study encountered computational difficulties with respect to the application of branch and bound techniques to solve Forest Utilisation Management Problems, the author recommends that branch and bound techniques be used in the solution of small-scale FUMPs having, say, fewer than 15 000 variables, of which fewer than 100 are integer.

- (e) It is recommended that consideration be given to solving Forest Utilisation Management Problems using network codes since, as indicated in subsection 3.5.2, computational advantage over traditional LP techniques may ensue.
3. (a) Model I or model II formulation may be used to model management activities within alternatives for management units. Neither formulation can be considered best under all circumstances. Generally, model I allows greater flexibility in detailing operations that can occur on a management unit because the area of the unit remains intact over the planning horizon. Model II makes it easier to impose area regulatory constraints, because areas harvested and stocked are readily available as part of the constraint structure.
- (b) Model II shows advantage for larger-scale aggregated problems in that it involves the specification of fewer variables.
 - (c) Mechanisms for capacity introduction which require integer variables can be either interpolating or non-interpolating. It is recommended that the interpolating version be used only when the duration of model periods is small in relation to the rotation lengths of the crop types being considered for harvest. The co-ordination of the timing of harvest, and utilisation of that harvest can influence the yield that can be drawn from management units. The interpolating version has obvious implications for harvest and utilisation of roundwood. To restrict the magnitude of yield anomalies that may arise, the model

period duration should be kept small in relation to crop-type rotation lengths.

4. (a) Expression of Forest Utilisation Management Problems as mathematical programming problems will probably grow in response to the question, "what action should be taken with respect to forestry production and processing?". The role of mathematical programming is that of a vehicle, albeit limited, for providing answers.
- (b) Network codes and mixed integer linear programs are particularly useful extensions of linear programming for representing Forest Utilisation Management Problems. This study shows formulation and construction can be via a number of alternative methods. These techniques should be regarded as optimal-seeking rather than optimal, *per se*, because of the errors manifest in the modelling process.
- (c) Finally, it is recommended that mathematical programming techniques be applied to both Forest Management Problems and Forest Utilisation Management Problems, with the proviso that the way of applying such techniques should remain open to debate.

REFERENCES

- Allison, B.J., Farquhar, G., Kane, W., Newell, K. and Sewell, W. 1979 "N.Z. Forest Products Ltd. Forest Information System Review" Proceedings N.Z. Forest Service F.R.I. Symposium No.20, pp 48-76.
- Apple, R.L. 1982 "An analysis of FORPLAN" Masterate thesis. Univ. of Minnesota, College of Forestry. 72 pp.
- Baird, F.T. and Whyte, A.G.D. 1982 "An interactive approach to modelling a forest sector" Proceedings, Operational Research Society of New Zealand, Aug. 1982, pp 61-9.
- Balas, E. 1967 "Discrete programming by the filter method" Operations Research, Vol. 15(5) : 915-57.
- Balinski, M.L. 1965 "Integer programming : methods, uses, computation" Management Science, Vol. 12(3) : 253-313.
- Bazaraa, M.S. and Jarvis, J.J. 1977 "Linear programming and network flows" Wiley, New York, 565 pp.
- Benders, J.F. 1962 "Partitioning procedures for solving mixed variable programming problems" Numerische Mathematik, Vol. 4 : 238-52.
- Berck, P. and Bible, T. 1984 "Solving and interpreting large-scale harvest scheduling problems by duality and decomposition" Forest Science Vol. 30(1) : 173-82.
- Boulding, K.E. and Spivey, W.A. 1960, Editors, "Linear programming and the theory of the firm" Macmillan, New Jersey, 227 pp.
- Brearley, A.L., Mitra, G. and Williams, H.P. 1975 "Analysis of mathematical programming problems prior to applying the simplex algorithm" Mathematical Programming, Vol 8(1) : 54-83.

- Burroughs Corporation 1976 "Model development language and report writer (MODELER)" User's Manual. No. 1094950. Detroit, Michigan.
- Burroughs Corporation 1975 "Tempo mathematical programming system" User's Manual. No. AA576436, Detroit, Michigan.
- Choi, J. and Nagumo, H. 1984 "A computerised planning system for a private forest" Proceedings, IUFRO Symposium on Forest Management Planning and Managerial Economics, Univ. of Tokyo, Oct. 1984, pp 332-42.
- Clutter, J.L. 1968 "Max-million : a computerized forest management planning system" School of Forest Resources, Univ. of Georgia.
- Clutter, J.L., Fortson, J.C., Pienaar, L.V., Brister, G.H. and Bailey, R.L. 1983 "Timber management : a quantitative approach" Wiley, New York, 333 pp.
- Dallenbach, H.G., George, J.A. and McNickle, D. 1983 "Introduction of operations research techniques" 2nd Ed. Allyn and Bacon, Massachusetts, 711 pp.
- Dantzig, G.B. and Wolfe, P. 1960 "Decomposition principle for linear programs" Operations Research, Vol. 8(1) : 101-11.
- Fourer, R. 1983 "Modelling languages versus matrix generators for linear programming" ACM Trans. Math. Software Vol. 9(2) : 143-83.
- Garcia, O. 1981 "IFS, An interactive forest simulator for long range planning" N.Z. Jl. Forestry Science 11(1) : 8-22.
- Garcia, O. 1984 "FOLPI, A forestry-oriented linear programming interpreter" Proceedings IUFRO Symposium on Forest Management Planning and Managerial Economics, Univ. of Tokyo, Oct. 1984, pp 293-305.
- Gass, S.I. "Linear programming" 4th Ed. McGraw Hill, New York, ¹⁹⁷¹ 406 pp.

- Glover, F. and Klingman, D. 1981 "The simplex SON algorithm for LP/embedded network problems" Mathematical Programming Study 15 : 148-76.
- Gomory, R.E. 1958 "Essentials of an algorithm for integer solutions to linear programs" Bulletin of American Mathematical Society, Vol. 64.
- Hadley, G. 1961 "Linear algebra" Addison Wesley, Reading Massachusetts, 290 pp.
- Hadley, G. 1962 "Linear programming" Addison Wesley, Reading Massachusetts, 484 pp.
- Hadley, G. 1964 "Nonlinear and dynamic programming" Addison Wesley, Reading Massachusetts, 484 pp.
- Hillier, F.S. and Lieberman, G.J. 1974 "Operations research" 2nd Ed. Holden-Day, San Francisco, 800 pp.
- Hitchcock, F.L. 1941 "Distribution of a product from several sources to numerous localities" Jl. Mathematical Physics, Vol. 20.
- Hoganson, H.M. and Rose, D.W. 1984 "A simulation approach for optimal timber management scheduling" Forest Science, Vol. 30(1) : 220-38.
- Intriligator, M.D. 1971 "Mathematical optimisation and economic theory" Prentice Hall, New Jersey, 508 pp.
- Johnson, K.D., Jones, D.B. and Kent, B.M. 1980 "Forest planning model (FORPLAN) : user's guide and operations manual" (draft copy) USDA For. Serv., Land Management Planning, Fort Collins.
- Johnson, K.N. and Scheurman, H.L. 1977 "Techniques for prescribing optimal timber harvest and investments under different objectives - discussion and synthesis" Forest Science monograph 18, 31 pp.
- Koene, J. 1983 "Minimal cost flow in processing networks, a primal approach" Centre for Mathematics and Computer Science Amsterdam, CWI tract, 157 pp.

- Koopmans, T.C. 1949 "Optimum utilisation of the transport system" *Econometrica*, Vol. 17, supplement.
- Land, A.H. and Doig, A.G. 1960 "An automatic method for solving discrete programming problems" *Econometrica* Vol. 28 : 497-520.
- Lasdon, L.S. 1970 "Optimisation theory for large systems" Macmillan series in Operations Research, New York, 523 pp.
- Leak, W.B. 1964 "Estimating maximum allowable timber yields by linear programming" USDA For. Serv., Res. paper NE-17.
- Littschwager, J.M. and Tcheng, T.H. 1967 "Solution of a large-scale forestry scheduling problem by linear programming decomposition" *Jl. Forestry*, Vol. 65(9) : 644-46.
- Luenberger, D.G. 1973 "Introduction to linear and non-linear programming" Addison Wesley, Reading Massachusetts, 356 pp.
- Mood, A.M., Graybill, F.A. and Boes, D.C. 1974 "Introduction to the theory of statistics" International Student Ed., 3rd Ed., McGraw Hill London, 564 pp.
- Murty, K.G. 1976 "Linear and combinational programming" Wiley, New York, 567 pp.
- Navon, D.I. 1971 "Timber Ram a long range planning method for commercial timber lands under multiple-use management" USDA For. Serv., Res. paper PNW-70, 22 pp.
- Navon, D.I. and McConnen, R.J. 1967 "Evaluating forest management policies by parametric linear programming" USDA For. Serv., Res. paper PSW-42.
- Paine, D.W.M. 1966 "Analysis of a forest management situation by linear programming" *Australian Forestry* Vol. 30(4) : 293-303.

- Rainville, E.D. and Bedient, P.E. 1969 "Elementary differential equations" Macmillan, New Jersey, 466 pp.
- Richardson, S.D. 1985 "Beyond the fringe" Proceedings, A.N.Z.I.F. Conference, Hobart, Tasmania, May 1985, pp 11-5.
- Ritter, K. 1967 "A decomposition method for linear programming problems with coupling constraints and variables" Mathematics Research Center, Univ. of Wisconsin, Rep. 739.
- Robinson, D.F. and Foulds, L.R. 1980 "Digraph theory and techniques" Gordon and Breach, New York, 256 pp.
- Rose, D.W. 1984 "A critical review of mathematical programming in forest resource planning" Proceedings, IUFRO Symposium on Forest Management Planning and Managerial Economics, Univ. of Tokyo, Oct. 1984, pp 306-17.
- Shirley, J.W. 1978 "Forest modelling : New Zealand Forest Service experience" Proceedings, F.R.I. Symposium No.20, Ref. 25, pp 391-400.
- Shirley, J.W. 1984 "Average yield of radiata pine in New Zealand state forests" N.Z. Jl. Forestry Vol. 29(1) : 143-44.
- Smyth, D.W. 1985 "Exporting roundwood (logs) : an analyst's appraisal of the New Zealand situation" Masterate report, School of Forestry, Univ. of Canterbury, N.Z., 215 pp.
- Suzuki, T. 1984 "The Gentan Probability, a model for the improvement of the normal forest concept and of forest planning" Proceedings, IUFRO Symposium on Forest Management Planning and Managerial Economics, Univ. of Tokyo, Oct. 1984, pp 12-24.
- Taha, H.A. 1976 "Operations research" 2nd Ed., Collier Macmillan London, 648 pp.

- Tcheng, T.H. 1966 "Scheduling of a large forestry-cutting problem by linear programming decomposition"
Ph.D. Thesis, Univ. of Iowa.
- Vaney, J.C. 1981 "Sawmill recovery and wood quality studies on pruned radiata pine trees from Mangatu and Patunamu State forests" N.Z.F.S. UDD Report No. 76.
- Wagner, H.M. 1975 "Principles of operations research with applications to managerial decisions" 2nd Ed., Prentice Hall, New Jersey, 1039 pp.
- Walker, B.A. 1982 "The role of quantitative modelling in forest management planning" Masterate thesis, Univ. of Minnesota, College of Forestry, 63 pp.
- Walsh, G.R. 1975 "Methods of Optimisation" Wiley, New York, 200 pp.
- Wardle, P.A. 1965 "Forest management and operational research : a linear programming study" Ser. B. Management Science, Vol.(11) 260-70.
- Ware, G.O. and Clutter, T.L. 1971 "A mathematical programming system for management of industrial forests" Forest Science Vol. 17(4) : 428-45.
- Whiteside, I. and Manley, B. 1985 "Log grading : F.R.I.'s answer to new crop log size problems" Forest Industries (N.Z.) Mar., 1985, pp 27-32.
- Whyte, A.G.D. 1984 "Management modelling of the forest sector in New Zealand" Proceedings, IUFRO Symposium on Forest Management Planning and Managerial Economics, Univ. of Tokyo, Oct. 1984, pp 25-34.
- Williams, H.P. 1978 "Model building in mathematical programming" Wiley, New York, 330 pp.
- Wilson, R.J. 1979 "Introduction to graph theory" 2nd Ed., Longman, London, 163 pp.

Zehna, P.W. 1970 "Probability distributions and Statistics"

Allyn and Bacon, Boston, 622 pp.

Appendix 2.1 Deterministic stand level volume considerations assume the yield at time t for a crop type can be represented by the function (1)

$$Y_t = f(t) \quad (1)$$

The Current Annual Increment may be defined as the derivative of the yield function (2). Foresters approximate this using non-central differences of the form (3) where the time increment is usually taken to be a year.

$$\frac{dY_t}{dt} = f'(t) \quad (2)$$

$$\frac{dY_t}{dt} = \frac{f(t+\Delta t) - f(t)}{\Delta t} \quad (3)$$

The maximum CAI occurs when growth is fastest. This follows from (4) and the observation that, for any rotation, growth initially increases then decreases. Any observed stationary point is a maximum because of this,

$$\frac{d}{dt} f'(t) = 0 \quad \text{or} \quad \frac{d^2y}{dt^2} = 0$$

The Mean Annual Increment (MAI) is defined by (5) to be the average volume produced by the rotation of time t .

$$MAI_t = \frac{f(t)}{t} \quad (5)$$

A stationary point for MAI requires that the CAI equals the value of MAI as (6) shows. This value is a maximum because $f(t)$ increases then decreases, which means that CAI and MAI increase then decrease as functions of time.

$$\begin{aligned} \frac{d}{dt} \left(\frac{f(t)}{t} \right) = 0 \quad \text{or} \quad \frac{t f'(t) - f(t)}{t^2} = 0 \\ \text{or} \quad f'(t) = \frac{f(t)}{t} \end{aligned} \quad (6)$$

Appendix 2.4.1 Model I Forest Management Problem

The model of Ware and Clutter (1971) is as follows.

$$\begin{aligned}
 &\text{maximise}_x \quad \sum_{i=1}^s \sum_{k=1}^m D_{ik} x_{ik} \\
 &\text{s.t.} \quad \sum_{i=1}^s \sum_{k=1}^m z_{ijk} x_{ik} \geq l_j \quad \forall j \\
 &\quad \quad \sum_{i=1}^s \sum_{k=1}^m z_{ijk} x_{ik} \leq f_j \quad \forall j \\
 &\quad \quad \sum_{i=1}^s \sum_{k=1}^m y_{ijk} x_{ik} \geq b_j \quad \forall j \\
 &\quad \quad \sum_{i=1}^s \sum_{k=1}^m y_{ijk} x_{ik} \leq c_j \quad \forall j \\
 &\quad \quad \sum_{k=1}^m x_{ik} = 1 \quad \forall i \\
 &\quad \quad x_{ik} \geq 0 \quad \forall i, j
 \end{aligned}$$

where

- Y_{ijk} is the yield of cutting unit i in period j under management alternative k ;
- D_{ik} is the total present value of cutting unit i if assigned to management alternative k ;
- z_{ijk} are units of area of cutting unit i regenerated in period j under management alternative k ;
- x_{ik} is the proportion of cutting unit i assigned to management alternative k .

Appendix 4.1 The Robinson mapping procedure.

The material contained in this appendix is included by permission of Dr. D.F. Robinson of the University of Canterbury Mathematics Department. It consists of excerpts from a paper prepared by Dr. Robinson entitled "Forward and Backward Functions of a Multi-Stage Process", which is, at the time of writing, pending publication. Dr. Robinson prepared this paper in response to a description of the problem being given by the author of this study.

Consider an n -stage production process represented by a (resource) digraph D . The vertex set V of D is partitioned into $n+1$ subsets V_0, V_1, \dots, V_n , for the various stages. Every arc of D joins a vertex in a set V_i to a vertex in V_{i+1} . The set V_0 consists of sources, and V_n of sinks.

We are given a set $U \subseteq V_0$ of raw materials. We seek sets X_0, X_1, \dots, X_n , subsets of V_0, V_1, \dots, V_n such that

1. $X_0 \subseteq U$
2. If $x_i \in X_i$, $1 \leq i \leq n$
then all precursors of x_i are in X_{i-1} ;
3. If $x_i \in X_i$, $0 \leq i \leq n-1$
then at least one successor of x_i is in X_{i+1} .

As $X_i = \{\emptyset\}$ is always a solution, we add

4. X_n is the largest possible.

We define two functions, ϕ (forward) and β (backward). Each maps sets to sets. Function ϕ maps subsets of V_i to subsets of V_{i-1} for $0 \leq i \leq n-1$, by the rule that if $X \subseteq V_i$,

$$\phi(X) = \{v; v \in V_{i-1}, \text{ all precursors of } v \text{ are in } X\}.$$

Function β maps subsets of V_i to subsets of V_{i-1} for $1 \leq i \leq n$,

by the rule that if $X \subseteq V_i$,

$$\beta(X) = \{u : u \in V_{i-1}, \text{ at least one successor of } u \text{ is in } X\}.$$

$\beta(X)$ thus consists of all the precursors of vertices in X .

The solution we would like to our problem is thus a sequence of sets X_0, X_1, \dots, X_n such that

1. $X_0 \subseteq U$
2. $X_i = \phi(X_{i-1})$ for $1 \leq i \leq n$
3. $X_i = \beta(X_{i+1})$ for $0 \leq i \leq n-1$
4. X_n is the largest possible.

Although these two sets of conditions seem the same at first sight, they are not identical, for in the first form 2, does not require that X_i be the largest possible set all of whose precursors are in X_{i-1} , as does the second form. We shall satisfy the first form, but not the second.

INITIAL THEOREMS

We now establish four theorems, which, for the purposes of later argument we will label P1 to P4. Since we will be dealing with sequences of ϕ and β , we shorten the notation by writing ϕX for $\phi(X)$ and βX for $\beta(X)$. Similarly, $\beta\phi X$ means $\beta(\phi(X))$.

- P1. If $W \subseteq X \subseteq V_i$ for $0 \leq i \leq n-1$
then $\phi W \subseteq \phi X$

Proof: Let $z \in \phi W$. Then all precursors of z are in W . But $W \subseteq X$, so all precursors of z are in X . Hence $z \in \phi X$. Thus $\phi W \subseteq \phi X$.

- P2. If $Z \subseteq Y \subseteq V_i$ for $1 \leq i \leq n$
then $\beta Z \subseteq \beta Y$

Proof: Let $w \in \beta Z$. Then w is a precursor of some $z \in Z$. as $Z \subseteq Y$, $z \in Y$ and w is a precursor of some

member of Y . Hence $w \in \beta Y$ and $\beta Z \subseteq \beta Y$.

P.3 If $X \subseteq V_i$, for $0 \leq i \leq n-1$
then $\beta\phi X \subseteq X$.

Proof: Let $x \in \beta\phi X$

Then x is a precursor of some $y \in \phi X$.

But as $y \in \phi X$, all precursors of y are in X
so $x \in X$.

Thus $\beta\phi X \subseteq X$.

P.4 If $Y \subseteq V_i$, for $1 \leq i \leq n$
then $Y \subseteq \phi\beta Y$.

Proof: Let $y \in Y$

Then all precursors of y are in βY

But $\phi\beta Y$ consists of those vertices all of
whose precursors are in βY .

Hence $Y \subseteq \phi\beta Y$.

We need one more property, which we take as a definition.
This will be necessary since on occasion the empty set will
be encountered.

P.5 $\phi\{\phi\} = \beta\{\phi\} = \{\phi\}$.

FURTHER DEVELOPMENT

Having established P2 to P5, we now take these as
axioms and prove the remaining theorems on that basis, without
reference to the initial problem. These theorems thus apply
to any system in which P1 to P5 can be established. We
retain as part of our structure the sets V_0 to V_n , though
interpretations of the axioms in which these have no
significance are presumably possible.

Theorem 6. If $X \subseteq V_i$ for $1 \leq i \leq n-1$
then $\beta\phi X \subseteq \phi\beta X$.

Proof : From P3 $\beta\phi X \subseteq X$

From P4 $X \subseteq \phi\beta X$

Hence $\beta\phi X \subseteq \phi\beta X$.

Theorem 7. If $X \subseteq V_i$ for $0 \leq i \leq n-1$
then $\phi\beta\phi X = \phi X$

Proof: By P3, $\beta\phi X \subseteq X$, so by P1,

$$\phi\beta\phi X \subseteq \phi X. \quad (1)$$

By P4, for any set $Y \subseteq V_{i+1}$,

$$Y \subseteq \phi\beta Y$$

Letting $Y = \phi X$

$$\phi X \subseteq \phi\beta\phi X \quad (2)$$

Combining (1) and (2) gives $\phi\beta\phi X = \phi X$.

Theorem 8. If $Y \subseteq V_i$ for $1 \leq i \leq n$
then $\beta\phi\beta Y = \beta Y$.

Proof: By P4, $Y \subseteq \phi\beta Y$, so by P2,

$$\beta Y \subseteq \beta\phi\beta Y \quad (1)$$

By P3, for any set $X \subseteq V_{i-1}$,

$$\beta\phi X \subseteq X.$$

Letting $X = \beta Y$,

$$\beta\phi\beta Y \subseteq \beta Y \quad (2)$$

Combining (1) and (2) gives $\beta\phi\beta Y = \beta Y$.

We are interested in sets formed from U by sequences of ϕ 's and β 's in any order, with the proviso that at no time do we go outside the bounds $0 \leq i \leq n$. We can check this does not happen by setting up a counter i , measuring the excess of the number of ϕ operations over β operations to the right of the marker at any point in the sequence. This gives the appropriate suffix for the set at that point.

We have only to ensure that $0 \leq i \leq n$ at all stages. For example, if $n=5$, the sequence $\phi\phi\beta\beta\phi\phi\beta\phi\phi\phi U$ can be marked by

$$\begin{array}{cccccccccccc} \phi & \phi & \beta & \beta & \phi & \phi & \beta & \phi & \phi & \phi & U \\ i & 4 & 3 & 2 & 3 & 4 & 3 & 2 & 3 & 2 & 1 & 0 \end{array} .$$

There is a way to avoid all reference to the limits on i ; this is to create for all negative integers i a copy of V_0 , with corresponding sets under the copying also corresponding under both ϕ and β , and achieving the same thing for $i > n$ by making copies of V_n . However, the results below will not necessarily hold in the form given here. We will use Γ, Δ to represent finite sequences of ϕ 's and β 's in any order, subject to the constraints just mentioned on the excess of ϕ 's over β 's to the right of any point in the sequence as a whole. We use these to give rules for shortening such sequences. Always, we operate on U , a subset of V_0 .

Theorem 9. If $\Gamma\phi\beta\phi\Delta U$ is valid, so is $\Gamma\phi\Delta U$, and

$$\Gamma\phi\beta\phi\Delta U = \Gamma\phi\Delta U.$$

Proof: If $\phi\beta\phi\Delta$ is valid, $\Delta U \subseteq V_i$

for some i , $0 \leq i \leq n-1$.

Then $\phi\Delta U \subseteq V_{i+1}$, $\beta\phi\Delta U \subseteq V_i$,

$\phi\beta\phi\Delta U \subseteq V_{i+1}$. Therefore throughout Γ , the set marker has the same value in $\Gamma\phi\Delta U$ as in $\Gamma\phi\beta\phi\Delta U$. Thus if one is valid, so is the other.

To show that they are equal, put $X = \Delta U$ in theorem 7, then

$$\phi\beta\phi\Delta U = \phi\Delta U.$$

Applying Γ to both sides yields the desired result:

$$\Gamma\phi\beta\phi\Delta U = \Gamma\phi\Delta U.$$

Theorem 10. If $\Gamma\beta\phi\beta\Delta U$ is valid so is $\Gamma\beta\Delta U$, and

$$\Gamma\beta\phi\beta\Delta U = \Gamma\beta\Delta U.$$

Proof: The validity is shown essentially as in theorem 9. Equality is shown by putting $Y = \Delta U$ in theorem 8:

$$\beta\phi\beta\Delta U = \beta\Delta U.$$

$$\text{Thus } \Gamma\beta\phi\beta\Delta U = \Gamma\beta\Delta U.$$

By means of theorems 9 and 10, we can shorten any sequence in which a ϕ has β on both sides, or a β has ϕ on both sides.

Theorem 11. If $\Gamma\beta\phi\Delta U$ is valid, so is $\Gamma\Delta U$, and $\Gamma\beta\phi\Delta U \subseteq \Gamma\Delta U$.

Proof: The validity is demonstrated as for theorem 9. The containment is shown by putting $X = \Delta U$ in P3.

Theorem 12. If $\Gamma\phi\beta\Delta U$ is valid, so is $\Gamma\Delta U$, and $\Gamma\phi\beta\Delta U \supseteq \Gamma\Delta U$.

Proof: The validity is demonstrated as for theorem 9. The containment is shown by putting $Y = \Delta U$ in P4.

Theorem 13. If ΓU is valid, then $\Gamma U \subseteq \phi^i U$
for some i , $0 \leq i \leq n$.

Proof: The significance of the i is that

$$\Gamma U \subseteq V_i.$$

If ΓU is valid, then the right most, first permormed, member of Γ must be ϕ .

Either Γ consists only of ϕ 's or there is a β in Γ with a ϕ immediately to the right of it. We write

$$\Gamma = \Gamma_1\beta\phi\Gamma_2$$

(in which Γ_1 or Γ_2 may be empty). Then by theorem 11

$$\Gamma_1\beta\phi\Gamma_2 U \subseteq \Gamma_1\Gamma_2 U.$$

We write $\Gamma_1\Gamma_2 = \Gamma'$.

Either $\Gamma' = \phi^i$ or we can again cancel a $\beta\phi$ combination.

As the initial sequence was finite in length, and we can always shorten the sequence so long as any β 's remain, we eventually reach $\Gamma U \subseteq \phi^i U$ as required.

Theorem 14. If ΓU is valid, then either

$$\Gamma U \supseteq \beta^{n-1}\phi^n U \text{ for some } i \quad 0 \leq i \leq n-1$$

$$\text{or } \Gamma U = \phi^n U$$

Proof: The sequence Γ must be one of three forms:

$$(1) \Gamma = \phi^i \text{ for some } i, \quad 0 \leq i \leq n$$

$$(2) \Gamma = \beta^k \phi^m \text{ for some } k, m, \quad 0 < k \leq m \leq n$$

(3) Γ is a mixture of ϕ 's and β 's with at least one ϕ to the left of some β .

In the last case, let $\Gamma = \Gamma_1\phi\beta\Gamma_2$,

then by theorem 12,

$$\Gamma_1\phi\beta\Gamma_2 U \supseteq \Gamma_1\Gamma_2 U$$

If $\Gamma' = \Gamma_1\Gamma_2$ is still in form (3), we repeat

the operation until, as must eventually

happen, either form (1) or form (2) is reached.

Thus for any ΓU , we have either

$$\Gamma U \supseteq \phi^i U \quad \text{or}$$

$$\Gamma U \supseteq \beta^k \phi^m U.$$

In order to attain the format of the theorem

we insert $\beta\phi$ as often as is necessary using

theorem 11.

$$\beta^k \phi^m U \supseteq \beta^k \beta \phi \phi^m U = \beta^{k+1} \phi^{m+1} U$$

As $k \leq m$ for validity, m will be increased to

n before k is. Indeed, k will be increased to n-i, where $\Gamma U \subseteq V_i$.

One special case deserves attention. If Γ is reduced precisely to ϕ^n , we have

$$\Gamma U \supseteq \phi^n U$$

But by theorem 13,

$$\Gamma U \subseteq \phi^n U,$$

so in this case we have the stronger result

$$\Gamma U = \phi^n U.$$

SOLUTION TO THE PROBLEM

Finally, let us return to our original problem. We have not found sets X_0, X_1, \dots, X_n as we desired. Given $U \subseteq V$, we find that we have all the resources required only for items in $U, \phi U, \phi^2 U, \dots, \phi^n U$. However, some initial resources or intermediate products may not be usable, as there is no guarantee that an item in any set X has any successors in ϕX . The set $\phi^n U$, on the other hand, does represent the maximal set of final products with the given resources, and $\beta^n \phi^n U$ is the subset of U consisting of those initial resources actually used to produce $\phi^n U$.

If we set

$$X_0 = \beta^n \phi^n U, X_1 = \beta^{n-1} \phi^n U, \dots, X_i = \beta^{n-i} \phi^n U, \dots, X_n = \phi^n U$$

we observe that these satisfy (1) to (4) in the original form, and are the maximal sets to do so. They do not necessarily satisfy the second form of conditions since there is no guarantee that $\phi X_{i-1} = X_i$.

Appendix 5.1.1(A)

This appendix indexes in file DB of appendix MYFILES the major data components by subject required in constructing the test problem. Two types of records are used in construction of the index: descriptive records, which identify a general area of required data (these records do not have line numbers); and indexing records, which indicate a set of data items followed by absolute line numbers in file DB where these data items can be found. A Nested classification is used in construction of the index. Generally descriptive records are provided and subsequent indexing records are indented.

current management units

land area	7-50
restocking crops	701-744
terrain specification	153-196
<i>production thinning</i>	
prior to planning horizon	51-44
during the planning horizon	216-259
<i>roading associated costs</i>	
prior to harvesting	344-363
<i>harvesting</i>	
hauler recovery costs	364-384
skidder recovery costs	385-406

yield

current (initial) crop	
unthinned stands	407-465
thinned stands	466-524
<i>restocked (subsequent) crop</i>	
unthinned stands	525-583
thinned stands	584-642
Administration establishment and tending	304-343

future management units

land area	95-113
land purchase cost	114-133
stocking crops	745-768
terrain specification	197-215
<i>production thinning</i>	
during the planning horizon	260-283
<i>roading associated costs</i>	
prior to afforestation	284-303
prior to harvesting	344-363

<i>harvesting</i>	
hauler recovery costs	364-384
skidder recovery costs	385-406
<i>yield</i>	
unthinned stands	643-671
thinned stands	672-700
Administration establishment and tending costs	304-343
bucking mechanism	973-1010
<i>management alternative generation mechanism</i>	
selection of block cutting patterns	841-894
block cutting patterns	769-840
<i>forest roundwood regulation</i>	
harvest volume smoothing	933-972
residual volume bounding and smoothing	895-932
<i>Transportation</i>	
<i>direction of arcs</i>	
forest to processing sites	1431-1475
forest to roundwood market	1476-1490
processing sites to lumber markets	1491-1541
<i>cost of arc flow</i>	
forest to processing sites	1031-1050
forest to roundwood market	1011-1030
processing sites to lumber markets	1542-1566
<i>Processing facility introduction</i>	
sites	1889-1889
facility type	1890-1890
facilities at sites	1401-1430
facility operating periods	1107-1155
facility introduction mechanism	1305-1400
	& 1195-1248

roundwood conversion at facilities

roundwood conversion costs	1156-1194
conversion product breakdown	1244-1304

roundwood market

market	1887-1887
roundwood product revenues	1567-1589

lumber and residue markets

markets	1888-1888
operating periods	1653-1673
lumber and residue product revenues	1590-1652
product set description	1634-1652

Appendix 5.1.1 (B)

This appendix is provided as a guide to interpreting the data base for the test problem (File DB in Appendix MYFILES). The material is largely sourced from the BURROUGHS "Model Development Language and Report Writer Manual" (1976). The syntax and semantics of the TABLE and LIST statements along with illustrative examples are presented.

Table statement.

A TABLE statement defines or reopens the definition of a table. It has the following form

$$\text{TABLE<identifier>} \left[\begin{array}{c} \{ \text{DIM1} \} \\ \{ \text{DIM2} \} \end{array} \right] \left[\begin{array}{c} \{ \text{ALPHA}[(\text{integer literal})] \} \\ \{ \text{INTEGER} \\ \text{REAL} \} \end{array} \right] \left[\text{EMPTY} \right] \left[\text{<data>} \right];$$

The identifier is the name of the table. The optional DIM1 indicates the table is one dimensional (c.f. one dimensional array). If DIM1 is omitted or DIM2 is specified, the table is two dimensional (c.f. two dimensional array). The optional ALPHA, INTEGER, or REAL, indicates whether the table contains alpha or numeric data. ALPHA indicates that all table data are 6-character alpha values (c.f. alpha array). ALPHA(n) indicates all table data are n-character alpha values ($n \leq 72$). INTEGER indicates the table contains numerical data and all data are truncated to an integer before being stored in the table. If the type is omitted, or if REAL is specified, the table contains single precision floating point numerical data. The option EMPTY indicates that all new table data locations are initialised to EMPTY (an attribute that can be tested for). If EMPTY is omitted, the default initialisation is zero for numerical tables and blank for alpha tables. The optional * indicates that table data follows in compile time readdata table formal (c.f., DATA statement in FORTRAN), the data

following must be terminated by a semicolon. Compile time readdata table format used * dimension name cards and table data cards.

* Dimension Name Cards.

A * dimension name card has the following form:

* <name>[<name>] . . .

These cards specify names in the first dimension list for one dimensional tables or the second dimension list for two dimensional tables. (c.f., one and two dimensional array indices). A <name> is an element or alpha literal.

Table Data Cards.

Table data cards have the following form

(a) One dimensional table:

{<datum>} . . .

(b) Two dimensional table:

<name> {<datum>} . . .

Table data cards specify data for a one dimensional table, (a) above, or data corresponding to a first dimension name in a two dimensional table, (b) above, (c.f. two dimensional array elements by row). The <name> is the first dimension name and is an element or an <alpha literal>. The <data> corresponds to the dimension names on the last * dimension name card. The datum is a numeric literal for numeric tables or an <element> or alpha literal for alpha tables.

Example 1 a one dimensional Table

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   TABULATION OF STOCKING CROP TYPES FOR FUTURE FOREST ESTATE
%
%   TABLE NAME
%   DESIGNATED BY                      INDEXED BY
%   STOCKCROP                          FOREST
%   STOCKCROP                          STOCKING PERIOD
%
%   TABLE DIMENSION NAME CARD
%   CROP TYPE
%
%   TABLE DATA CARD
%   CROP TYPE STOCKING INDICATOR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TABLE STOCKCROP00 DIMI REAL EMPTY
*      1      2      3      4      5
          1
;

```

The Table element 1 is here taken to indicate that the relevant crop-type (in this case 2) can be considered for stocking purposes for the forest indexed (0), and the stocking period indexed (0). Positions 1,3,4, and 5 are EMPTY and will have numerical value of zero if accessed.

Exampe 2 a two dimensional Table

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% TABULATION OF SKIDDER LOGGING COSTS
%
%   TABLE NAME
%   DESIGNATED BY           INDEXED BY
%   SKIDCOST                FOREST
%
%   TABLE DIMENSION NAME CARD
%   HARVEST AGE
%
%   TABLE DATA CARD
%   CROP TYPE
%   SKIDDER LOGGING COSTS IN $ PER CUBIC METRE RECOVERED
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TABLE SKIDCOST0 DIM2 REAL
*   15    20    25    30    35    40    45    50    55    60
1 10.60  7.40  6.40  4.90  4.20  4.00  4.00  3.75
2 15.00  8.20  7.00  6.60  5.40  5.00  4.60  4.35
3      15.00  8.20  7.30  6.90  6.60  6.40  6.00
;

```

The Table entries are indexed by crop-type (row) and age (column) and represent the skidder extraction costs (\$/m³) for logs in the crop-type age class combinations shown in forest 0. The elements of the table not containing entries would yield zero if accessed.

List Statement.

A list statement defines or reopens the definition of a list. It has the following form:

```
LIST<identifier> [CHECK] [ { SORTDOWN }
                        { SORTUP  } ] [*<data>];
```

The identifier is the name of the list. The optional CHECK indicates that an element is added to the list in the statement only if it is not already in the list. Duplicate elements are not added, duplicate elements already in the list

are not affected. The optional SORTDOWN or SORTUP indicates that at the end of the LIST statement, the entire list is to be rearranged with its elements sorted in descending or ascending order according to a machine dependent alpha collating sequence. The optional * indicates that the list elements follow in compile time readdata list format.

The compile time readdata list format is as follows:

$$\{ \langle \text{element} \rangle \quad [\langle \text{alpha literal} \rangle] \} \cdot \cdot \cdot$$

An element is a group of up to 72 characters which cannot include special characters or blanks. Each element can have optional text information associated with it, specified by the alpha literal. Data is free format with elements separated by at least one blank. An <element> can not be continued from one card-image to the next. If text is specified the <alpha literal> follows its associated element.

Example 3 a list with Text

```
LIST PERIOD * 0 "1986-90" 1 "1991-95" 2 "1996-00"
               3 "2001-05" 4 "2006-10" 5 "2011-15" ;
```

Both TABLE and LIST statements permit clauses which allow run time manipulation of table or list elements, additionally table and list references can be used in assignment statements and as operands or arguments within boolean and numeric expressions. A full description of these factors is given in the BURROUGHS "Model Development Language and Report Writer Manual" (1976).

Appendix 5.1.2

The table formats presented in this appendix describe report formats that can be used to present solution data resulting from runs of a model. The full report on the solution for the test problem uses similar table formats and is contained in file RP in Appendix MYFILES. The following symbolic forms are used in this appendix.

A sequence of one or more dollar signs is taken to be a string value such as a forest name, or facility name that is problem dependent and is provided during the report writing phase. Special uses of this symbol are as follows: the sequence \$\$\$\$-\$ is taken to be a time interval (e.g., 1986-90); repetition of this sequence is taken to be a set of adjoining time intervals (e.g., the following

\$\$\$-\$ \$\$\$\$-\$ \$\$\$\$-\$ could be taken to mean

1986-90 1991-95 1996-00); additionally the sequence \$\$\$\$ alone is taken to be the model base year (e.g., 1986). When a \$ sign is contained in a string enclosed by quotes then its actual meaning is inferred (e.g., "\$/ha" for dollars per hectare).

A sequence of one or more lower case letters d within a table body are used to represent a set of decimal digits which is taken to be a real number in exponential, decimal, or integer form if it has the following format
± dd...dE±dd, dd...d·d...d, dd...d respectively.

Report on Current Estate Management Units

\$\$\$\$\$\$\$\$\$\$

Tabulations for each Alternative may include

A summary showing standing recoverable volume

A summary showing harvested recoverable volume

Summary type	period stock	area altn m.u.	"\$/ha" altn	crop int	p. thin sub	Volume distribution by period				
						\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	
Stand						cu m	cu m	cu m	cu m	
skid	\$\$\$\$_\$\$	dddd.dd dddd.dd	ddddddddd	\$	\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d	
haul	\$\$\$\$_\$\$	dddd.dd dddd.dd	ddddddddd	\$	\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d	
Harvest						cu m	cu m	cu m	cu m	
skid	\$\$\$\$_\$\$	dddd.dd dddd.dd	ddddddddd	\$	\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d	
haul	\$\$\$\$_\$\$	dddd.dd dddd.dd	ddddddddd	\$	\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d	

Report on Future Estate Management Units

\$\$\$\$\$\$\$\$\$\$

Tabulations for each Alternative may include

- A Summary showing standing recoverable volume
- A Summary showing harvested recoverable volume

Summary	period	area		"\$/ha"		crop	Volume distribution by period			
type	stock	altn	m.u.	altn	m.u.	id pt	\$\$\$\$-\$\$	\$\$\$\$-\$\$	\$\$\$\$-\$\$	\$\$\$\$-\$\$
Stand				ha	ha		cu m	cu m	cu m	cu m
skid	\$\$\$\$-\$\$	dddd.dd	dddd.dd	ddddddddd	ddddddddd	\$ \$	ddddddd.d	ddddddd.d	ddddddd.d	ddddddd.d
haul	\$\$\$\$-\$\$	dddd.dd	dddd.dd	ddddddddd	ddddddddd	\$	ddddddd.d	ddddddd.d	ddddddd.d	ddddddd.d
Harvest		ha	ha				cu m	cu m	cu m	cu.m
skid	\$\$\$\$-\$\$	dddd.dd	dddd.dd	ddddddddd	ddddddddd	\$ \$	ddddddd.d	ddddddd.d	ddddddd.d	ddddddd.d
haul	\$\$\$\$-\$\$	dddd.dd	dddd.dd	ddddddddd	ddddddddd	\$	ddddddd.d	ddddddd.d	ddddddd.d	ddddddd.d

Resource Report on Forest

\$\$\$\$\$\$\$\$\$

Tabulations for each Resource may include

- An Area Summary showing hectares standing by time
- An Area Summary showing hectares harvested by time
- A Stand Summary showing standing volume by time
- A Stump Summary showing recoverable volume by time

			Resource distribution by period			
crop age			\$\$\$\$_\$	\$\$\$\$_\$	\$\$\$\$_\$	\$\$\$\$_\$
Stand area			ha	ha	ha	ha
skid	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
haul	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
Harvt area			ha	ha	ha	ha
skid	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
haul	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
Stand vol			cu m	cu m	cu m	cu m
skid	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
haul	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
Harvt vol			cu m	cu m	cu m	cu m
skid	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
haul	\$	dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d

Crop Log Class Report in Forest Estate

\$\$\$\$\$\$\$\$\$\$

Tabulations for each Crop Log class include

A Summary showing standing volume by crop log class

A Summary showing harvest volume by crop log class

Summary		Crop log class vol distbn by period			
Type	crop log	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$
		cu m	cu m	cu m	cu m
stand	\$ \$\$\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d
harvest	\$ \$\$\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d

Report on Residual Volume by Forest

Tabulations for each forest indicate

the distbn of residual volume by period
 - residual vol is taken to be standing
 vol in harvestable age classes

Forest identity code	_\$\$ cu m	_\$\$ cu m	_\$\$ cu m	_\$\$ cu m
\$	+ E+dd	+ E+dd	+ E+dd	+ E+dd

Report on Harvest Volume by Forest

Tabulations for each forest indicated

The distbn of harvest volume by period
 - harvest vol is taken to be harvested
 vol from harvestable age classes

Forest identity code	\$\$\$\$-\$\$ cu m	\$\$\$\$-\$\$ cu m	\$\$\$\$-\$\$ cu m	\$\$\$\$-\$\$ cu m
\$\$\$\$\$\$\$\$\$ \$	+ddddddE+dd	+ddddddE+dd	+ddddddE+dd	+ddddddE+dd

Forest Site Arc Report

Tabulated are Roundwood Volumes Shipped by Period

the arcs are in the direction forest to utilisation site
the following codes are employed in the tabulation

Forest (F)	Code	Site (S)	Code
=====	=====	=====	=====
\$\$\$\$\$\$\$\$\$\$	\$	\$\$\$\$\$\$\$\$\$\$	\$

arc			\$\$\$	\$\$\$_\$\$	\$\$\$_\$\$	\$\$\$_\$\$	\$\$\$_\$\$
F	S	crop log	arc cost	cu m	cu m	cu m	cu m
+	+	+	+	+	+	+	+
\$	\$	\$	\$\$\$ dddd.dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
+	+	+	+	+	+	+	+

1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

Tabulated is an operation period summary
for facilities at the specified site

Site Code	Plant Code	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$
\$	\$				

Site Capital Finance Report

Tabulated for facilities introduced at the specified site are

a statement of the monies necessary for the "introduction"
a refinancing scheme- payments due at start of the period

site code	plant code	intro' monies	repayment installments in N.Z. dollars			
			\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$
\$	\$	+ddddddE+dd	+ddddddE+dd	+ddddddE+dd	+ddddddE+dd	+ddddddE+dd

Facility Input Report

Tabulated are crop log roundwood volumes utilised by period

Site				Facility			
=====				=====			
\$\$\$\$\$\$\$\$\$\$				\$\$\$\$\$\$\$\$\$\$			

site plant				\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$
code code crop log				cu_m	cu_m	cu_m	cu_m

\$	\$	\$	\$\$\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d

[illegible]

Tabulated is a Summary of Product units produced by period

[illegible]

Source	Year	Country	Population	Age Group	Gender	Study Type
1	2000	USA	200,000	18-25	Male	Survey
2	2001	USA	150,000	26-35	Female	Survey
3	2002	USA	100,000	36-45	Male	Survey
4	2003	USA	50,000	46-55	Female	Survey
5	2004	USA	25,000	56-65	Male	Survey
6	2005	USA	12,500	66-75	Female	Survey
7	2006	USA	6,250	76-85	Male	Survey
8	2007	USA	3,125	86-95	Female	Survey
9	2008	USA	1,562	96-105	Male	Survey
10	2009	USA	781	106-115	Female	Survey
11	2010	USA	390	116-125	Male	Survey
12	2011	USA	195	126-135	Female	Survey
13	2012	USA	97	136-145	Male	Survey
14	2013	USA	48	146-155	Female	Survey
15	2014	USA	24	156-165	Male	Survey
16	2015	USA	12	166-175	Female	Survey
17	2016	USA	6	176-185	Male	Survey
18	2017	USA	3	186-195	Female	Survey
19	2018	USA	1	196-205	Male	Survey
20	2019	USA	0	206-215	Female	Survey
21	2020	USA	0	216-225	Male	Survey
22	2021	USA	0	226-235	Female	Survey
23	2022	USA	0	236-245	Male	Survey
24	2023	USA	0	246-255	Female	Survey
25	2024	USA	0	256-265	Male	Survey
26	2025	USA	0	266-275	Female	Survey
27	2026	USA	0	276-285	Male	Survey
28	2027	USA	0	286-295	Female	Survey
29	2028	USA	0	296-305	Male	Survey
30	2029	USA	0	306-315	Female	Survey
31	2030	USA	0	316-325	Male	Survey
32	2031	USA	0	326-335	Female	Survey
33	2032	USA	0	336-345	Male	Survey
34	2033	USA	0	346-355	Female	Survey
35	2034	USA	0	356-365	Male	Survey
36	2035	USA	0	366-375	Female	Survey
37	2036	USA	0	376-385	Male	Survey
38	2037	USA	0	386-395	Female	Survey
39	2038	USA	0	396-405	Male	Survey
40	2039	USA	0	406-415	Female	Survey
41	2040	USA	0	416-425	Male	Survey
42	2041	USA	0	426-435	Female	Survey
43	2042	USA	0	436-445	Male	Survey
44	2043	USA	0	446-455	Female	Survey
45	2044	USA	0	456-465	Male	Survey
46	2045	USA	0	466-475	Female	Survey
47	2046	USA	0	476-485	Male	Survey
48	2047	USA	0	486-495	Female	Survey
49	2048	USA	0	496-505	Male	Survey
50	2049	USA	0	506-515	Female	Survey
51	2050	USA	0	516-525	Male	Survey
52	2051	USA	0	526-535	Female	Survey
53	2052	USA	0	536-545	Male	Survey
54	2053	USA	0	546-555	Female	Survey
55	2054	USA	0	556-565	Male	Survey
56	2055	USA	0	566-575	Female	Survey
57	2056	USA	0	576-585	Male	Survey
58	2057	USA	0	586-595	Female	Survey
59	2058	USA	0	596-605	Male	Survey
60	2059	USA	0	606-615	Female	Survey
61	2060	USA	0			

\$\$\$\$\$\$\$\$\$

product	product	production distribution by period			
description	units	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$	\$\$\$\$_\$\$
\$\$\$\$\$\$\$\$\$\$\$\$	\$\$\$\$\$\$\$	dddddd.d	dddddd.d	dddddd.d	dddddd.d

Site Market Arc Report

Tabulated are product units shipped by Period

the arcs are in the direction utilisation site to market
the following codes are employed in the tabulation

Site	Code	Market	Code
====	====	=====	====
\$\$\$\$\$\$\$\$\$\$	\$	\$\$\$\$\$\$\$\$\$\$	\$

product	product	\$\$\$	product distribution by period			
description	units	arc cost	\$\$\$_\$\$	\$\$\$_\$\$	\$\$\$_\$\$	\$\$\$_\$\$
-----	-----	-----	-----	-----	-----	-----
\$\$\$\$\$\$\$\$\$\$\$\$	\$\$\$\$\$\$\$	dddd.dd	dddddd.d	dddddd.d	dddddd.d	dddddd.d
-----	-----	-----	-----	-----	-----	-----

Roundwood Market Report \$\$\$\$\$\$\$\$\$\$

Tabulated are crop-type log class roundwood vols sold by period

market				_	_	_	_
id	crop log	"\$/cu m"		cu_m	cu_m	cu_m	cu_m
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+
\$	\$	ddd.d		ddddddd.d	ddddddd.d	ddddddd.d	ddddddd.d
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+

Market Product Report

Tabulated are product units sold at the designated market by time

Market

=====

\$\$\$\$\$\$\$\$\$\$\$

market	product	product	\$\$\$ unit	product distribution by period
id	description	units	revenue	\$\$\$_\$ \$\$\$\$_\$ \$\$\$_\$ \$\$\$_\$
\$	\$\$\$\$\$\$\$\$\$\$\$	\$\$\$\$\$	dddd.dd	dddddd.d ddddddd.d ddddddd.d ddddddd.d

Appendix 5.1.3

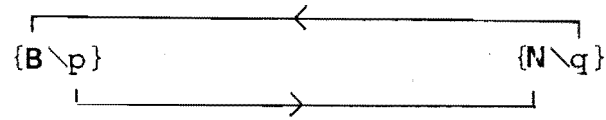
A Branch and Bound procedure to solve Mixed Integer Linear Programs (MILP's) requires an algorithm to solve (LP's) that arise during the fathoming of the problem (Dallenbach *et al.*, 1983). Fathoming is achieved by relaxation of integer constraints and separation of the problem into sub-problems, such that, every feasible solution to the original problem is a feasible solution to one and only one subproblem, and every feasible solution of every subproblem is a feasible solution to the original problem. Relaxation gives rise to linear programs, while separation gives rise to specification of constraints on activities that are required to be integer.

The constraints on activities required to be integer can be made to appear as bounded variable constraints, the motivation for doing this is that the size of the required basis is reduced. Thus, the simplex procedure with facility to cope with upper bounds is a suitable algorithm to solve LP's that arise during solution of MILP's. This algorithm, generally known as the simple upper bounding algorithm (SUB) differs from the standard simplex procedure in the way primal feasibility is maintained. Thus a different set of rules is used selecting the pivot element by the SUB algorithm (Gass, 1975; Luenberger, 1973). Once a pivot element has been identified, the operations to change the basis are entirely the same. Moreover, the condition to ensure that the basis is always non-singular is readily derived.

A basis transfer in the simplex procedure involves transfer of a vector, a_q say, from the set of non-basic variables to the basis, while a vector a_p say, is transferred from the set of basic variables to the non-basic set.¹ The

¹ Each iteration of the simplex or revised simplex procedure involves a basis transfer, whereas for the SUB procedure non-basic variables can change between their upper and lower bounds (and vice versa) without a basis transfer.

basic transfer may be illustrated as follows.



where

B = set of basic variables, N = set of non-basic variables.

Consider the basis update procedure, subsequent to the performing of pivot operations, i.e., vector q added and vector p deleted. The new basis corresponds to

$$\hat{B} = B + (a_q - a_p) e_p^T \quad (1)$$

where

a_p, a_q = column vectors of dimension $m \times 1$ representing the vector leaving and entering the basis respectively.

e_p^T = elementary row vector of dimension $1 \times m$ with a 1 in the p th position.

B = updated basis set after addition of a_q and deletion of a_p .

The inverse of \hat{B} is determined as follows

$$\begin{aligned} \hat{B}^{-1} &= [B + (a_q - a_p) e_p^T]^{-1} \\ &= [B (I + B^{-1} (a_q - a_p) e_p^T)]^{-1} \\ &= \left[I - \frac{B^{-1} (a_q - a_p) e_p^T}{1 + e_p^T B^{-1} (a_q - a_p)} \right] B^{-1} \end{aligned} \quad (2)$$

From (2), it is apparent that the new inverse exists provided

$$1 + e_p^T B^{-1} (a_q - a_p) \neq 0 \quad (3)$$

Evaluating the components in (3), $e_p^T B^{-1} a_p = 1$ since a_p is the basis vector, and $B^{-1} a_q = y_q$ the

representation of a_q in terms of the basis B , then $e_p^T B^{-1} a_q = y_{pq}$, the p^{th} component of y_q . Thus (3) then simplifies to (4) which states the inverse exists provided the pivot (y_{pq}) is non-zero (pivot selection rules always ensure this).

$$y_{pq} \neq 0 \quad (4)$$

The form (2) above is equivalent with more familiar forms of updating a tableau, namely, the product form of the inverse. This equivalence can be shown as follows.

$$\begin{bmatrix} 1 & -\frac{B^{-1}(a_q - a_p)e_p^T}{1 + e_p^T B^{-1}(a_q - a_p)} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} -y_{1q}/y_{pq} \\ -y_{2q}/y_{pq} \\ \vdots \\ (y_{pq} - 1)/y_{pq} \\ \vdots \\ -y_{mq}/y_{pq} \end{bmatrix}$$

\uparrow
 p^{th} column

$$= \begin{bmatrix} 1 & -y_{1q}/y_{pq} \\ & 1 & -y_{2q}/y_{pq} \\ & & \ddots \\ & & & -1/y_{pq} \\ & & & \vdots \\ & & & -y_{mq}/y_{pq} & 1 \end{bmatrix}$$

\uparrow
 p^{th} column

$$= [e_1, e_2, \dots, e_{p-1}, \eta, e_{p+1}, \dots, e_m]$$

$$= E \quad (5)$$

where

$$\eta = \left[-y_{1q}/y_{pq}, -y_{2q}/y_{pq}, \dots, -1/y_{pq}, \dots, -y_{mq}/y_{pq} \right]^T \quad (6)$$

Thus update of the inverse is achieved by pre-multiplication of the previous inverse B^{-1} by the elementary matrix E as in (7). This is the so called product form of the inverse (Hadley, 1962).

The vector in (6) is termed an Eta vector, only this vector and pointer information need be stored to reconstruct E , the elementary matrix in (5).

$$\hat{B}^{-1} = EB^{-1} \quad (7)$$

Linear programming algorithms that require the maintenance of a basis inverse may use different techniques to accomplish this. A common procedure is the product form of the inverse, adopted because of its compactness and speed by which multiplications by the inverse may be performed. Using this method, the inverse is decomposed to a sequence of matrix multiplications (viz., the required multiplications after k basis transfers assuming the initial basis was an identity matrix is shown in (8)). Procedures using the product form of the inverse must either store the sequence of elementary matrices in order to reconstruct the basis inverse (as in (8)), or store the Eta vectors (6) and pointer information to reconstruct the elementary matrices.

$$\hat{B}^{-1} = E_k E_{k-1} \dots E_1 \quad (8)$$

$$\hat{B}^{-1} = E_k E_{k-1} \dots E_1 B^{-1} \quad (9)$$

Should the sequence of elementary matrices in (8) become too long machine roundoff error can occur, to prevent this the current basis can be reinverted using an inversion

procedure (e.g., the TEMPO inversion procedure is called INVERT and is called by default by the TEMPO version of the revised simplex procedure called PRIMAL). Subsequent to the reinversion the formula (9) may be used to maintain the inverse of the current basis.

Appendix 5.2.1

The MODELER test programs shown in Figures 1 and 2 construct matrices of some dimension (i.e., $10^2 \times 10^2$), with each element being assigned the value 1. These programs differ only in the control of the statement used. The program in Figure 1 explicitly replicates a COLUMN statement using DO's. In Figure 2, control of the column statement is implicit. The MODELER boolean option CHECK may be SET or RESET. Only when CHECK is true is the order of generation preserved under all circumstances; e.g., it is not possible to split the definition of a row using a ROW statement when CHECK is false. The CPU and I/O times, tabulated in Tables 1 and 2, relate to the running of programs in Figures 1 and 2 and row-use generation programs (not shown). The programs were run in a time sharing environment on a BURROUGHS B6930 machine at Canterbury University, other users of the machine were absent at the time of execution.

```

$ SET LIST
%
% Modeler implicit column statement and clause generation
% test program
%
LIST ROWSET1      *    0    1    2    3    4    5    6    7    8    9;
LIST ROWSET2      *    0    1    2    3    4    5    6    7    8    9;
LIST COLSET1      *    0    1    2    3    4    5    6    7    8    9;
LIST COLSET2      *    0    1    2    3    4    5    6    7    8    9;
%
CHECK = TRUE;
%
COLUMN "C"&COLSET1&COLSET2
, "R"&ROWSET1&ROWSET2=1;
%
MODEL MATRIX PRINT;
%
EXIT;

```

Figure 2 : Modeler test program for generation of matrix with $10^2 \times 10^2$ elements with implicit control of both column statement and clause.

```

$ SET LIST
%
% Modeler explicit column statement and clause generation
% test program
%
LIST ROWSET1      *    0    1    2    3    4    5    6    7    8    9;
LIST ROWSET2      *    0    1    2    3    4    5    6    7    8    9;
LIST COLSET1      *    0    1    2    3    4    5    6    7    8    9;
LIST COLSET2      *    0    1    2    3    4    5    6    7    8    9;
%
CHECK = TRUE;
%
DO (COLSET1)
  DO (COLSET2)
    DO (ROWSET1)
      DO (ROWSET2)
        COLUMN "C"&COLSET1&COLSET2
        , "R"&ROWSET1&ROWSET2=1;
%
MODEL MATRIX PRINT;
%
EXIT;

```

Figure 1 : Modeler test program for generation of matrix with $10^2 \times 10^2$ elements with explicit control of both column statement and clause.

CHECK	row stmt		col stmt	
	CPU	I/O	CPU	I/O
true	70	3	210	179
false	61	8	239	169

Table 1 : Tabulated are CPU and I/O times in seconds for modeler programs generating a $10^2 \times 10^2$ matrix and assigning 1 to each element. Explicit replication of statement and clause were used.

CHECK	row stmt		col start	
	CPU	I/O	CPU	I/O
true	52	2	286	164
false	35	2	187	162

Table 2 : Tabulated are CPU and I/O times in seconds for modeler programs generating a $10^2 \times 10^2$ matrix and assigning 1 to each element. Implicit replication of statement and clause were used.